Final Report
Small Business Innovation Research Contract
Phase I

Optimizing Cockpit Display Configurations with a Genetic Algorithm System

Tica Technologies, Inc.
36 Hampshire Street
Cambridge, MA 02139

Dec. 1, 1994

Contract NAS2-14052
NASA Ames Research Center
Army Aeroflightdynamics Directorate
COTR: Barry Smith

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Dec. 1, 1994

3. REPORT TYPE AND DATES COVERED
Final report. 6/94-11/94

4. TITLE AND SUBTITLE
Optimizing Cockpit Display Configurations
with a Genetic Algorithm System

5. FUNDING NUMBERS
C (contract)
NAS2-14052

6. AUTHOR(S)
Lawrence Davis, Betsy Constantine,
Stuart Shieber, Joe Marks, Rebecca Hwa

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Tica Technologies, Inc.
36 Hampshire St.
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER
SBIR-I-94-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Army Aeroflightdynamics Directorate
co-monitored by NASA Ames Research Center
Moffett Field, CA 94035-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
No limitations on distribution

12b. DISTRIBUTION CODE
No Limitations

13. ABSTRACT *(Maximum 200 words)*

Our Phase I work resulted in a software prototype for MFD page organization, a technique for translating MFD page organizations into graph partitioning problems, and a genetic algorithm for carrying out the graph partitioning. We apply our system to a scenario taken from the Comanche attack helicopter task analysis. The system out-performs other published algorithms for graph partitioning.

14. SUBJECT TERMS
Cockpit design, MFD design, Comanche attack
helicopter, genetic algorithms, optimization

15. NUMBER OF PAGES
140

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

Table of Contents

Summary

This report describes the results of a Small Business Innovation Research project carried out from June 1 to November 30, 1994. The project's goal was to investigate the application of genetic algorithms to problems in aviation cockpit design.

The project team consisted of three members from Tica Technologies, Inc., two members from Harvard University, and one member from Harvard University and MERL.

The project had three goals: 1. the specification of the information requirements for a one-hour scenario taken from the full mission specification of the Comanche attack helicopter; 2. the assignment of display items to pages of a Multiple Function Display device with a genetic algorithm so that pilots could most effectively execute the mission as specified by the scenario; and 3. the creation of the Display Optimizer, a prototype software system demonstrating how a cockpit designer might use the system to explore the effects of various design decisions on the cockpit design.

Each of these goals was met. With regard to the first, the team extracted a challenging one-hour scenario from the Comanche mission specification. With regard to the second, the team designed a novel procedure for transforming the Multiple Function Display page organization problem into a problem called graph partitioning, and designed a genetic algorithm that out-performs all other published algorithms for partitioning graphs of the type that is relevant to our cockpit design problem. With regard to the third, the team produced and demonstrated the Display Optimizer, a software system that shows how the research we have done could be used by cockpit designers. We should like to note that in achieving these results, we have advanced the state of the art both in the evaluation of Multiple Function Display designs, and in graph theory.

We find these results to be quite encouraging, and recommend that the project be considered for Phase II funding. We should like to note that although our work has centered on cockpit design, the techniques we have developed are pertinent to any human-machine interface environment in which the interface involves multiple computer screens. For example, techniques of the sort we have developed in completing this phase of the project could be used in designing interfaces to Information Superhighway sites.

I. Introduction

This report describes the activities carried out by Tica Technologies, Inc., its subcontractors from Harvard University, and its collaborator from MERL on a Phase I Small Business Innovation Research project entitled "Cockpit Design with Genetic Algorithms." The work was sponsored by the Army Aeroflightdynamics Directorate, and was administered under contract NAS2-14052 by NASA Ames Research Center during the six-month period from June 1 1994-November 30 1994.

The following sections of this report are organized as follows.

Section II: Project Staff

In section II of this report we describe Tica Technologies, Inc. and its participation in the project, our subcontractors and their participation, and our collaborator and his participation.

Section III: The Problem to be Solved

In section III of this report we discuss the general cockpit design problem and the part of it that we addressed in our project, the MFD page organization problem.

Section IV: Project Activities by Task

The tasks as broken down in our Statement of Work include identifying the constraints that a cockpit configuration system should satisfy and choosing a representative subset of those constraints to be included in our Phase I efforts; designing the Phase I mission scenario and its performance metrics; specifying the interface between the scenario and the MIDAS modules that must be used to run the scenario; implementing the scenario and a minimal version of MIDAS; designing and implementing a genetic algorithm to configure cockpits for the Task I scenario; and reporting on the potential of the genetic algorithm for cockpit configuration, based on its performance on this scenario. We provide a detailed discussion of our progress on each of these tasks in section IV of this report.

Section V: Description of Results

In section V of this report we characterize the results of our research. The results include a working software prototype that shows how a designer might interface with our system, a useful algorithm for transforming MFD page organization problems into graph partitioning problems, and a genetic algorithm for solving graph partitioning problems effectively.

Section VI: Summary of Results

In section VI of this report we characterize the results of our investigations and we note that the project has demonstrated that evolutionary algorithms show great promise for assisting human designers in producing effective cockpit designs.

Section VII: Future Work

In section VII we describe a number of improvements and enhancements that could be made to the current Display Optimizer. We conclude by recommending that the project be considered for subsequent funding.

Bibliography.

We include a bibliography of references cited.

Addenda. The appendices to this report include a general description of genetic algorithms, a description of the file format that constitutes the Display Optimizer's interface with MIDAS, a description of the mission scenario that we used as a test case, graphical examples of the application of a transformation procedure that we devised for this project, documentation of the results produced when the Display Optimizer is applied to the test scenario, a description of the Display Optimizer and its interface, and a precise characterization of a genetic algorithm for assigning display items to pages of a Multiple Function Device.

We impose no restrictions on the information in this report, but would like to note that patent disclosures have been filed on the genetic algorithm described in Appendix G, developed before this project began, and on the transformation procedure described below, developed in the course of our research on this project.

II. Project Staff

Tica Technologies, Inc. was particularly delighted to win this contract from NASA Ames Research Center under the United States Army's Small Business Innovation Research program, because each of us has been involved before in one or more of the problem areas that we worked on in this project. Below we describe our six team members and their areas of expertise.

Dr. Lawrence Davis, President of Tica Technologies Inc., and Principal Investigator for the project, has been involved with the MIDAS project as a consultant for nine years. Dr. Davis is generally recognized as the world's leading authority on genetic algorithm optimization. Dr. Davis has been implementing applications of genetic algorithm technology for twelve years. He is author/editor of the *Handbook of Genetic Algorithms*, the only existing text on genetic algorithm applications. He edited *Genetic Algorithms and Simulated Annealing*, and he founded Tica Technologies, Inc. (then Tica Associates) in 1990 to specialize in genetic algorithm applications. Dr. Davis writes a column on genetic algorithm applications for a newsletter on advanced technology, and is the author or co-author of forty papers, including more than fifteen on genetic algorithm applications. Dr. Davis' body of work includes several projects directly related to the present task: development of genetic algorithms for semiconductor layout under physical constraints at Texas Instruments (Smith and Davis 1985), development of genetic algorithms for telecommunication network design under performance and cost constraints at Bolt Beranek and Newman Inc. (Davis and Coombs 1987; Coombs and Davis 1987), and development of genetic algorithms for survivable network design under performance, cost, and algorithmic constraints at Tica Associates in conjunction with U. S. West (Davis, Cox, Orvosh, and Qiu 1993). Dr. Davis has also carried out pioneering research work showing how repair strategies and constraint-based mutation strategies can enhance and speed up genetic algorithm performance in domains like the present one (Davis 1993; Cox, Davis, and Qiu 1991; Orvosh and Davis 1993).

In addition to his genetic algorithm work, while at Bolt Beranek and Newman Inc in 1985 Dr. Davis implemented the first version of the MIDAS system. During the past eight years, Dr. Davis has continued to provide consultation and software for the MIDAS system, first at Bolt Beranek and Newman, and since 1990 at Tica Technologies, Inc.

Dr. Davis directed the Phase I project, concentrating most heavily on the topics of cockpit configuration research issues related to genetic algorithms, enhancement of the current genetic algorithm technology to support the requirements of the present project, and analysis of the system performance for the final report.

Dr. Betsy Constantine, a Vice President of Tica Technologies, Inc., is a psychologist with eleven years' experience in the area of human-computer interface. From 1982 to 1984 she managed a group working on human factors engineering of speech recognition systems. From 1984 to 1988 she contributed to and managed many projects as a senior staff member in the AI Application Center of Arthur D. Little, Inc. For this work she performed task analyses, knowledge modeling and system design for AI-based systems, experience which was directly applicable to the task of establishing the constraints and scenario for Phase I. Following her employment at Arthur D. Little, Inc., Dr. Constantine was a neural network researcher and developer of training materials on neural networks and other advanced technologies.

Important to the success of our project is the fact that from 1990 to 1992 Dr. Constantine served as Task Manager for Sterling Software in the Computational Human Engineering Research Office at Ames Research Center. In this position she managed the MIDAS software developers during a time when the MIDAS architecture was completely overhauled. She is thoroughly familiar with the design and implementation of MIDAS and with the principles that have guided its development. She also has a good working knowledge of the details of the portions of MIDAS with which the Display Optimizer will interact.

Dr. Constantine's role in the project has concentrated on scenario development, constraint specification, and cockpit configuration issues related to genetic algorithm optimization.

James Kelly, an employee of Tica Technologies, Inc. is noted for his work on systems that simulate human performance, including a system that replicates the selection of tax returns for audit by expert IRS auditors. This system is expected to be the expert system with greatest return of any expert system ever produced, owing to the magnitude of the auditing effort of the Internal Revenue Service. Mr. Kelly is also noted for his work on databases and interfaces. In this project, he implemented the interface to the Display Optimizer and its database containing design information.

Dr. Stuart Shieber is a professor in the Computer Science Department at Harvard University. Dr. Shieber has been an active researcher on human-machine interface issues during the past four years. His selection as a Presidential Faculty Fellow was partly due to his work in this field. This award, given yearly to only thirty researchers in all fields of science in the United States, recognizes the level of the abilities that Dr. Shieber brought to our project.

Dr. Shieber's role in the project lay in the areas of optimization algorithm design, in the transformation of the problem into a graph partitioning problem, and in supervising the work of Ms. Hwa.

Rebecca Hwa is a graduate student in the Computer Science department of Harvard University who has specialized in topics related to graphical presentation of information to humans and the automation of information source layout so that human performance is enhanced. Ms. Hwa's role in the project centered on the conversion of the problem into a graph partitioning problem, in the design and implementation of the optimization algorithm used in the prototype software system, and in the testing of the algorithms used in the prototype system.

Dr. Joe Marks is a researcher at MERL (Mitsubishi Electric Research Laboratory) and is an adjunct professor in the Computer Science department at Harvard University. One of Dr. Marks' research specialties is the effective presentation of information to humans. Dr. Marks' prior research has advanced the state of the art in several areas of graphical presentation and layout.

Although Dr. Marks' participation in this project was not funded by the project, the project was extremely fortunate to have him as a team member. He was a significant contributor to the project in designing the transformation algorithm and in collaborating on the design of the genetic algorithm described in Appendix G of this report.

The accumulated experience of these six project members and the level of their expertise allowed us to create a system that goes beyond the current state of the art in assisting cockpit designers to organize MFD information sources, and in optimizing the organization of information for designers of other types of human-machine interfaces.

As we will show, the interactions of the skills possessed by these six project staff members have led us to significant results. We believe those results will have impact on problems beyond those of cockpit design.

III. The Problem to be Solved

The cockpit is an information-rich environment in which a pilot's failure to receive or understand a critical item of information can result in aircraft loss and/or loss of human life. In addition, efficient pilot performance can result in significant savings in aircraft fuel and maintenance costs. For these reasons, designing and appropriately configuring cockpit information display devices is an essential part of the process of design of a modern aircraft.

Traditionally, cockpit information displays (for instance, altimeters and oil pressure gauges) have been dedicated, hard-wired devices that have taken up all of the cockpit display area. Scanning such devices and integrating the information displayed on them consumes a good deal of a pilot's sensory and cognitive resources. Cockpit layout design decisions for these devices have been primarily concerned with: 1) *physical constraints* on display locations (*e.g.*, no overlaps), and 2) *human factors constraints* based on research in human performance (such as stimulus-response compatibility, as described in Andre, Wickens, and Goldwasser 1990 and Vincow and Wickens 1992). A great deal of human engineering expertise has been developed over the past fifty years to deal with the problem of positioning dedicated displays in a cockpit.

The cockpit configuration problem is currently changing radically, however, and many of the design principles developed for individual, dedicated display devices are no longer applicable to the modern "glass" cockpit. As aircraft systems become more complex, crew members must be aware of increasing amounts of information of growing complexity. As a result, the amount of information that must be provided to the cockpit crew has increased greatly, far exceeding the capacity of dedicated displays. Newer cockpit information display devices (for instance, Multiple Function Displays, or MFDs) take up less of the cockpit display area by concentrating the output of multiple information sources on a single, multiple-paged display device. However, the problem of maintaining appropriate access to relevant information in the cockpit has changed dramatically from one of scanning fixed, dedicated displays at appropriate intervals to one of managing the content of the available display space in time. Some of the pilot's concerns with this type of display device have to do with knowing how to move from page to page of the display to acquire information, and remembering what pages of the display contain the necessary information when unexpected situations arise.

New engineering principles and new rules of thumb are needed to guide the layout and design of such display devices. These design principles are critical to pilot success, and many of them cannot be gained from prior experience, since existing aircraft do not use display devices of the type to be installed in future aircraft. How are such design principles to be developed and tested?

One approach to this problem has been the development of the Man-machine Integration Design and Analysis System (MIDAS) by the Computational Human Engineering Research Office (CHERO) at NASA's Ames Research Center under the Army-NASA Aircrew/Aircraft Integration (A3I) Program. MIDAS has been developed as a framework in which to explore solutions to crewstation design problems such as that of finding the most effective information display configuration. To accomplish this goal, MIDAS includes an unparalleled dynamic computer simulation capability which can model aircrew performance in a specified crewstation under simulated mission conditions. The MIDAS mission simulation system includes models of human behavior and performance which interact with models of crew station equipment, aircraft dynamics, and environment to dynamically generate a mission scenario as it unfolds, providing the user with analyses of critical areas of human operator performance, such as visual perception, decision making, and workload.

To use MIDAS, the user specifies 1) the physical and functional design and configuration of crew station equipment, displays and controls, 2) a mission scenario, with a route of flight and waypoints, and activities to be performed by operators interacting with the cockpit design elements and 3) operator characteristics, such as size, and selected cognitive characteristics, such as scheduling strategy or memory decay rate. Given this input, MIDAS provides the crewstation designer with human factors analyses, such as reach and visibility, as well as mission and operator performance measures obtained by computing a simulated mission scenario.

A crewstation designer may try several configurations of cockpit displays and rerun the simulation to determine the effect of a new cockpit configuration on mission performance. However, there are many constraints on the configuration and a trial-and-error approach is inefficient and possibly ineffective. The new types of constraints on cockpit display layout and configuration create a critical need for the development of a new optimization system that is capable of balancing all these complex and conflicting constraints to produce a cockpit configuration that effectively supports the expected missions.

We have created a software system, the *Display Optimizer*, that can interact with MIDAS (or any other models of pilot performance) to improve our ability to design and configure information displays in aircraft cockpits. One critical component of the Display Optimizer is a *genetic algorithm*, an "evolutionary" approach to design that improves on current designs by "evolving" better and better ones. Genetic algorithms are optimization techniques that have recently been applied to a diverse array of real-world problems. We provide an overview of the genetic algorithm technology in Appendix A. In this report, we explain how we applied this technology to the problem of MFD page organization, in order to "evolve" components of the cockpit layout so as to interact well with our model of pilot performance.

IV. Project Activities by Task

In this section of the report, we describe those activities carried out to successfully complete each of the six tasks specified in the Statement of Work.

Task 1: Identify the full range of constraints that a cockpit configuration system should satisfy, and choose a representative subset of those constraints to use in a Phase I test scenario.

This task was completed during the two-day project kickoff meeting in June of 1994. We approached the two components of the task in reverse order. First, we settled on the type of scenario to implement. In consultation with Barry Smith, the COTR, and with human factors experts from NASA Ames, it was decided to shift the emphasis in Phase I of the SBIR project from two-dimensional layout of information sources in the cockpit (the topic suggested in our proposal) to the allocation of information sources to pages of an aircraft's Multiple Function Display (MFD) pages.

This change in emphasis was adopted for two reasons. First, the NASA Ames human factors experts believe that the problem of organizing information sources on MFD pages is a harder problem than that of positioning information sources in the cockpit. Second, since MFDs are a relatively recent development in cockpit design, more research is needed to improve their effectiveness.

Having decided to make MFD page organization the primary topic of Phase I, we then settled on a domain. The aircraft with the most interesting MFD configuration problem known to the group attending our kickoff meeting is the Comanche attack helicopter. Based on the recommendations of the Ames human factors experts and the COTR, we determined that Phase I would produce a software system that would organize information sources on MFD pages for the Comanche helicopter, using segments of the Comanche mission description as input to the design process.

After we had settled on a domain, we dealt with the question of the constraints the Phase I system was to handle. Some constraints that are relevant to the problem of two-dimensional layout are not relevant to the problem of organization of information sources on MFD pages. Examples are stimulus compatibility, symmetrical layout, and so forth—constraints on the actual arrangement of objects on the page, rather than on the assignment of objects to pages. On the other hand, some constraints are highly relevant to MFD page allocation. Examples that were highlighted by participants in our kickoff meeting include:

> *efficiency of traversal*; sequences of information source accesses by the pilot should be efficient.
>
> *accessibility of emergency procedures;* emergency procedures should be easily accessed from MFD pages devoted to normal operating procedures.
>
> *compactness of design;* the number of MFD pages should be minimized where possible.
>
> *accessibility of the home page;* the entry page of the MFD should be easily accessible from any of the procedures.
>
> *constraints on size;* no page should have information sources assigned to it that cannot be fit on the page.
>
> *criticality and frequency;* the design of the MFD pages should take into account the criticality of the tasks that involve interaction with the MFD, and it should take into account the frequency with which the MFD information sources are accessed.

Some of these constraints on MFD page assignment are incompatible. What is required is a way to describe the importance of each constraint on the design so that an automated design system can produce a good design, while adjudicating among the constraints. The sense of the kickoff meeting was that a Phase I design system that could take these constraints into account when organizing information sources into MFD pages would have accomplished a significant task.

There were some constraints that were discussed that were not included in the Phase I project, including these:

> *number of cross-references;* some designers believe that humans cannot remember more than a few cross-links in navigating the MFD page space. Accordingly, some designers feel that MFD designs should be created with minimal numbers of cross-links.

minimal depth; some designers believe that no MFD page should be more than three accesses away from the home page.

For the purposes of our research in Phase I, at the recommendation of COTR Barry Smith we settled on the initial set of constraints listed above. The number of cross-references was not used as a constraint because quantifiable data describing the effect of cross-references on pilot mental models of the MFD are not known. It will be a simple matter to add this constraint if the performance of the system with respect to the others is judged promising. Minimum depth was not used as a constraint because it was felt that if this constraint were realistic, it would be a by-product of a design created with attention paid to other features. Both these constraints can be added in a straightforward way to the system we have produced.

Task 2. Design the Phase I scenario and its performance metrics.

In our kickoff meeting, we determined that a subset of the mission description for the Comanche helicopter would comprise the Phase I scenario. It was Dr. Constantine's responsibility to extract such a scenario from the Comanche mission description. It was important to produce a scenario that would be short enough to be treated with the resources available under a Phase I SBIR contract, and that would be long enough and complex enough to provide data that would be of interest to observers of the project.

Dr. Constantine spent September and October working with mission description materials furnished by NASA Ames in order to find subsets of the mission description that would exercise our system's ability to solve the MFD page organization problem.

Dr. Constantine began with the Pilot-Vehicle Interface Mechanization Specification (PVIMS) for the Comanche helicopter, which includes the results of a task analysis for several mission scenarios. She chose one mission scenario from this document. The scenario is described in section 2.2.2.4.1.2, "RAH-66 Comanche Armed Reconnaissance Mission Timeline." It was not immediately clear how to derive pilot accesses of MFD information sources from this document. Dr. Constantine proceeded as follows.

She analyzed the first hour of this scenario by referring to the Phase-Segment specification given in the scenario. Consulting the Phase-Segment analyses referenced in this portion of the mission scenario, she listed in chronological order the functions allocated to the pilot and copilot against a timeline with a brief summary of the mission events. She determined that we would consider only the copilot MFD for purposes of the Phase I study, since of the two pilots, the copilot's activities tend to be more oriented toward the MFD. Accordingly, she studied the Function Analysis section of the PVIMS in which each function is broken down into pilot and copilot tasks at the button press level. From careful study of the function analysis for each function, she determined what information was required for the copilot to perform each function and in what order the information was to be accessed.

In this way, Dr. Constantine was able to generate a list of functions that would be performed by the copilot during the first hour of the selected mission scenario. This list of functions is shown in pages C-1 and C-2 of Appendix C. There are two points to note concerning this function list. First, certain functions such as Perform External Voice Communication that do not involve information that would ordinarily be displayed on an MFD were omitted from our function list, since they do not impact the MFD design. Second, some functions, notably Perform Navigation, are continuous functions and thus require virtually continuous access to the required display items. In its Phase I implementation, the Display Optimizer does not handle continuous display requirements. (We discuss this issue further in the Results section of this report.)

The list of functions in Table 1 was used to generate input data for our test problem. Associated with each function is a sequence of display items that must be accessed by the copilot in order to perform the function. What we wish to do is facilitate access to the required display items so that the work the copilot needs to do to obtain the required information is reduced or minimized.

After deriving sequences of display item accesses for standard procedures occurring in the first hour of the scenario, Dr. Constantine derived similar sequences for twelve functions that are emergency procedures. The emergency procedures are shown page C-3 of Appendix C. These functions were included in the function list specifically to show that the optimization algorithm could handle such procedures, minimizing the number of responses required to access the information necessary in carrying out an emergency procedure, regardless of when in the scenario the emergency occurred.

The sources we used for this analysis and some of our results are shown in Appendix C. The portion of the mission scenario we used for our analysis appears on pages C-4 through C-13. Two examples of the Phase-Segment analyses are shown on pages C-14 and C-15. The lists of functions we obtained for the pilot and copilot for the first hour of the mission are shown on pages C-16 through C-19. Two examples of the PVIMS Function Analysis are shown on pages C-20 through C-22.

Dr. Constantine's analysis of the MFD-related functions occurring in the first hour of the Comanche mission, together with sequences of display item accesses required to execute the 12 additional emergency procedures, constituted the scenario that was the principal object of study in Phase I. From these sequences of display item accesses we would proceed to investigate techniques for clustering display items on pages so that the effort required of the copilot to access information when executing a function was minimized.

Task 3. Specify the interface between the scenario and the subset of the MIDAS modules that must be used to run the scenario.

This task was begun during the two-day project kickoff meeting in June, and was completed in July and August by Ms. Hwa in Cambridge. Because the emphasis of the project had shifted from two-dimensional layout of graphical display elements to the organization of MFD pages, requirements for interfacing the Display Optimizer with MIDAS were greatly reduced. MIDAS stores and manipulates a good deal of information about the size, shape, and physical characteristics of information display devices. This information can be given to the Display Optimizer in file form; in fact, Dr. Constantine's analysis of the scenario descriptions produced such an input to our system. MIDAS has not been configured to include Comanche attack helicopter missions, and so there was no need to go beyond a file interface format as an interface to MIDAS. Ms. Hwa prepared the specification of a file-based interface that appears in Appendix B. Our studies used files in this format to communicate between the designer module of our software system and the optimization module. MIDAS could incorporate a designer/user interface much like that in our prototype and generate the file for input to the optimization module.

Task 4. Implement the scenario and a minimal version of MIDAS at the Tica Technologies, Inc. offices.

This task was completed during the final two months of the project. Since MIDAS does not at present contain Comanche attack helicopter scenarios, the minimal version of MIDAS that we produced consisted simply of the creation of files describing mission activities in the format specified in Appendix B. Some of the material in these files was computer-generated by our scenario design module, with results like those MIDAS would produce if it contained Comanche scenario information. Some of the material was hand-designed by Dr. Constantine to mimic the information that MIDAS would write into a scenario description file if it did include Comanche scenario information.

Because it would not have impacted the project to implement a version of MIDAS for the purposes of the Phase I study, in consultation with COTR Barry Smith we determined instead to produce a prototype of the Display Optimizer to demonstrate how our system might assist a cockpit designer in developing MFD page structures, given an understanding of an aircraft's mission requirements.

Task 5. Design and implement a genetic algorithm to configure cockpits, using the scenario of Task 4 as the genetic algorithm's evaluation function.

This task was the sole activity for Dr. Shieber, Dr. Marks, and Ms. Hwa. It was the principal activity for Dr. Davis and Mr. Kelly. The problem of converting a mission scenario description of the form described in Appendix B into an organization of display items on MFD pages is a difficult one. The approach we used is innovative and ingenious, and was devised by Dr. Shieber, Dr. Marks, and Ms. Hwa.

Our approach begins by translating descriptions of scenarios in the format described in Appendix B into descriptions of a different sort—descriptions of graphs and their nodes and links. When this transformation is accomplished, the MFD page organization problem is formally equivalent to the problem of partitioning the nodes of the graph into sets such that the total size of each set is less than a maximum size, and such that the aggregate weights of the links between sets is minimized. When the graph partitioning problem is solved using one of the techniques detailed below, the result can be translated back into an organization of display items on MFD pages that satisfies the constraints of the design.

A detailed discussion of our work on Task 5 is included in Section V. A good deal of time was spent devising the transformation procedure, through which a page organization problem was transformed into a problem equivalent in relevant aspects that a genetic algorithm could be used to solve. A good deal of time was also spent customizing the genetic algorithm so that it was effective in the MFD page organization domain.

Task 6. Report on the potential of the genetic algorithm for cockpit configuration, based on its performance on the page organization problem.

We have reported on our progress in three ways: a mid-project review meeting held at NASA Ames in October, a final meeting held in mid-November, and this report. We discuss the results of our work and detail our conclusions and recommendations for future work in the following sections of this report.

V. Description of Results

In this section of the report, we provide a detailed discussion of two major results of our work: the development of transformation and optimization algorithms for solving the MFD page organization problem, and the application of those algorithms to our test problems

A. The Display Optimization Algorithm

Our display optimization algorithm has two parts. The first, a transformation algorithm, transforms an MFD page organization problem into a graph partitioning problem. The second, an optimization algorithm, is a genetic algorithm that solves the associated graph partitioning problem. Before discussing these two algorithms, we provide some terminology.

1. Terminology

In what follows, we shall use the following terminology. A *graph* is a structure consisting of *nodes*, each of which has a size, and *links*, each of which passes in a single direction between two nodes, and each of which has a weight. Roughly put, we shall create a graph that represents the various paths that pilots must follow from display item to display item in the MFD when they execute the functions that make up a mission. The size of each node is the relative size that its corresponding display item takes up on an MFD page. The weight on each link represents the combined frequency and importance of the functions that require passing from the page containing the node representing the first display item to the page containing the node representing the second display item.

2. Transforming an MFD page organization problem into a graph partitioning problem

The MFD page organization problem and graphs as we have just described them do not at first glance appear to have much in common. Seeing that they share common formal properties was a nontrivial, significant, and enabling insight on the part of the Harvard/MERL members of the project team.

Once we understand that the page organization problem can be viewed as a kind of graph partitioning problem, we can then apply a variety of known technologies, including genetic algorithms, to the translated form of the page organization problem in order to find high-quality organizations of display items into pages. In order to understand how this comes about and in order to understand the strengths and weaknesses of our approach, we shall explain in some detail what it means to transform a problem of MFD page design into a graph problem with the same characteristics. Below we describe the steps required to carry out the transformation along with the intuitions behind each step.

The following steps describe the process of transformation of an MFD organization problem into a graph partitioning problem that has the relevant characteristics of the MFD organization problem. To aid in understanding the process, we show for each of the first six steps what happens to a simple illustrative example. Visual illustrations of the example and its state after the first six steps of the transformation process are contained in Appendix D.

For the purposes of our example, we assume that the size of any link from a node on one page to a node on a different page is 1 unit, and 0 units otherwise. We assume that the maximum size capacity of a page is 7 units. Finally, we assume that the display items to be assigned to MFD pages and their associated area sizes are as follows: A 4, B 3, C 4, D 4, E 2, F 2, and G 1. This is the information that would be entered by a designer using our prototype interface, and corresponds to the notion that large displays take up more space on a page than does a single-line display.

A simplifying assumption here is that display item sizes are additive. This assumption is not necessarily correct. It may be possible to have two display items whose sizes sum to less than the size of a page, but that are configured so that they cannot both be arranged on the page. These features of the problem have to do with what we call *bin packing* below. We recommend that such constraints be addressed in a further phase of this research, in which a bin packing module is added to the system.

Step 1: Create graph nodes

Given this information we begin the transformation process by creating nodes for the graph in the following way:

> *Node creation process: For each display item, create a node in the graph with the same name as the display item. Associate with this node a number that is the display item's size.*

For our example, carrying out this process creates seven graph nodes, A-G. The result is shown on page D-1 in Appendix D, where nodes A-G have been added and their sizes have been associated with their corresponding nodes. The intuition here is that display items will form the resting points, or nodes, of the graph. Moving from one display item to another in carrying out a function will correspond to moving from one node in the graph to another along a link. The sizes of each node will become important later when we form groups of nodes that will go on a single page.

Step 2: Introduce links for sequences of display item accesses

The designer has provided us with sequences of accesses to display items that are necessary to perform the functions in the scenario. Each sequence has associated with it two numbers: a *frequency rating* and a *criticality rating*. For each sequence in the set of sequences, we carry out the following procedure for each pair of display items in the sequence:

> *Creation of links for sequences: If there is no link in the graph from the first item to the second item, add a link with a weight equal to the product of the sequence's frequency rating times its criticality rating. If there is a link in the graph from the first item to the second item, increase the weight on that link by the product of the sequence's frequency rating times its criticality rating.*

This process creates links or increments the weight of existing links in the graph that connect the display items in the sequence. To illustrate this step in the transformation for our example, we assume that there are only two sequences of display item accesses. The first consists of accesses of the nodes B, A, F, E, and D, in that order. We suppose that the frequency of this sequence is 5 and the criticality of this sequence is 1. Page D-2 shows the result of adding links to the graph for this sequence. Note that each link has an arrow showing the direction of movement in the graph. The links go from B to A, from A to F, and so on. Note also that each link has weight 5, which is the product of this sequence's frequency times its criticality.

Now let us consider the addition of links to our example graph for a second sequence with frequency 3 and criticality 2. This sequence consists of accesses of the display items C, A, F, and G, in that order. Page D-3 shows the result of adding links to the graph for this sequence. Note that a new link has been added from node C to node A, with weight 6 (frequency 3 times criticality 2). Note that a link already existed from node A to node F, and so the weight of that link has been incremented by 6 to 11 (the old weight of 5 plus the new weight of 6). Finally, a new link with weight 6 has been added from node F to node G.

The intuition behind this procedure is that movement from display item to display item by a pilot corresponds to movement from node to node in the graph. If separate functions require movement between the same nodes, we recognize this by summing the effects of these movements. For our purposes, moving between two nodes with a single sequence of criticality 3 and frequency 1 is just as important as moving between those two nodes in three sequences of criticality 1 and frequency 1. In both cases, what we are representing with a link weight is the importance of the load imposed on the copilot by moving from the first display item to the second. Solving the MFD design problem means minimizing this load.

We would like to note that we have chosen to amalgamate the two distinct measurements of frequency and criticality by forming their product. Any other technique for combining them could be incorporated here with no change to the algorithm. It would also be possible to consider them as distinct quantities, so that each link had an associated frequency and criticality rating.

Step 3: Add links for always-accessible nodes

One of the constraints to be satisfied by our system was the requirement that certain procedures, typically emergency procedures, be readily accessible no matter what functions were being executed by the pilot at the time when the emergencies arose. To satisfy this constraint, we introduce the notion of an *always-accessible node*.

An always-accessible node is a node representing a display item that should be accessible from any page in the MFD. It would be impossible to have every such node be instantly accessible from every page if there were a substantial number of such nodes, since buttons to access the pages on which those nodes reside would consume more than the total space on other pages. Hence, part of the MFD design problem involves constructing a pattern of display item access that allows ready processing of normal display sequences, while allowing rapid access of emergency sequences. This part of the problem is represented in our graph as follows:

> *Creation of links for always-accessible nodes: For each always-accessible node, create a link to that node from every other node in the graph.*

Let us assume that the first sequence in our example was an emergency procedure. We represent the requirement that this procedure must be accessible at any time by adding links to it from every other node in the graph. The result is shown on page D-4. Node B, the first node in the emergency procedure, now has links added from every other node. In cases in which such links already existed, no new link is added. Instead, the weights on such links are incremented by a new parameter value equal to the importance of accessing this emergency procedure. In our example, this parameter value was 50.

The intuition behind this procedure is that emergency procedures should be readily accessible. In our approach, we have satisfied this constraint by adding links to the first node of each emergency procedure from all the other nodes in the graph. We would like to note several points about this procedure.

First, the procedure as we have described it is not exactly correct. Emergency procedures will not in general need to be reachable from every other function, and the weights on the links that reach them will not in general be equal. An example of an emergency procedure with some links not needed is the emergency procedure for an impending midair collision, which need not be accessible from the sequence of activities carried out during the preflight checklist. Our goal in Phase I was not to represent in realistic fashion each constraint on the MFD design. This was not possible given project resources. Our goal was to show how a given constraint could be plausibly satisfied. In the case of emergency procedures, a good deal of knowledge-based information could be added to the problem database by a designer. This information would include the conditions under which emergency procedures were relevant, and the importance of initiating those procedures quickly. This information could be translated in a straightforward way into the addition of fewer links in the graph than we are currently adding, and into the addition of a variety of more meaningful weights on the links that are added. We wish to stress that nothing in what follows depends on how one decides what links to add or how to weight them.

The second point we wish to note is that the addition of these links to the graph guarantee that emergency procedures will be taken into account when the MFD pages are organized, and it provides a way for us to measure the adequacy with which their incorporation has proceeded. Current design practice does not explicitly integrate emergency procedure performance with standard mission performance. For example, the mission scenario from which the designer works does not contain explicit information concerning the probability of encountering emergencies of various types during the mission, and there appears to be no quantitative method for measuring an MFD design's success in handling emergency situations. We believe that the transformation procedure we are describing here provides such a method, and that this may be one of its most valuable features.

Step 4: Create Conceptual Clusters

One type of problem specification concerns conceptual clusters—groups of display items that the designer wishes to group together. In this step of the transformation, the requirement that conceptually related display items be placed together is added to the graph.

It is important to note that display items in a conceptual cluster may not ultimately be placed on the same MFD page. The items in a cluster may occupy more area than a single page can hold, and efficient traversal of the MFD pages may require that some clusters be broken up so that others can be maintained while keeping the total number of MFD pages low. Thus, the optimization system to be described below is given an *incentive* to place conceptually clustered display items together by the procedure we are describing, but it is not required to place clustered items together if breaking them up creates a design that is better for other reasons.

The specification of a conceptual cluster includes the names of the nodes that form the cluster, and a weight, or importance, to be attached to keeping the cluster together. The following procedure adds information about conceptual clusters to the graph:

> *Addition of links for conceptual clusters: For each conceptual cluster with importance w, consider each pair of nodes in the cluster. If links exist in either direction between the nodes, increment the weight of the existing links by w. If either of the two links does not exist, add it and assign it weight w.*

For purposes of our example, let us assume that display items E, F, and G form a conceptual cluster with an importance of 2. We see on page D-5 that a link with weight 2 has been created from E to F, and the existing link from F to E has had its weight incremented by 2 from 3 to 5. We see also that a link has been added from G to F, and the existing link from F to G has had its weight incremented by 2. We see finally that links have been added from E to G and from G to E. As we shall note later, these links provide a strong incentive for optimization algorithms to group E, F, and G on a single page, but they do not make such a grouping obligatory.

The intuition behind this step of the transformation is that conceptual clustering information should bias the design, but should not completely constrain it.

Step 5: Add Stepping Stone Nodes

As we shall explain below, the transformation process is designed to weight the links in our graph so that each link represents the effort expended by a pilot in moving from one display device to another if the two display devices are not on the same page. The process as it stands has not done that, because each pair of nodes has a single link between them, whereas it may not be possible to move from one node to another directly. For cases in which an intermediate page must be accessed in order to pass from one display item to another, the weights on the links in our graph are incorrect. To be correct, the graph should contain a node for each intermediate page, and a new link from each intermediate page to the next page on the path to the second display item. This step of the transformation process takes this possibility into account. It proceeds as follows:

> *Addition of stepping stone nodes: For every link l in the graph, introduce a new node n and change l so that l points to the new node. Introduce a new link from n to the node that l originally pointed to.*

This procedure introduces new nodes corresponding to pages that might be accessed if the MFD is organized so that display devices are separated by an intermediate page. The stepping stone nodes represent the possibility of an intermediate page being visited when navigating from one display item to another in the MFD. In practice, this rarely occurs, and as we shall see in our examples, most of the stepping stone nodes will not appear in a typical MFD design. One should note that if the design is sufficiently complicated that there is a possibility of two pages being accessed between a pair of sequence nodes, then this procedure should be modified to add two stepping stone nodes on every link.

Let us observe how this procedure is applied to our example. In order that the graph not be overly cluttered, we apply this step to the graph shown on page D-3 (our example graph after step 2 had been applied) rather than to the full graph derived above. The result is shown on page D-6, where new nodes have been introduced in each existing link.

Step 6 : Add a home page

A common feature of MFD organization is the existence of a *home page*, a page where interaction with the MFD begins, and to which the pilot returns frequently before going to a new sequence of display item accesses. The home page should be easily accessible when functions are completed, and it should provide access to the sequences of functions to be performed. The following procedure can be used to create a home page:

> *Add home page: Add a new node (the home page node). Add a link from this node to the first node of every function sequence. Add a link from the end of every function sequence to this node. Add stepping stone nodes to each of the links just added. Set the size of the home page node to 0. Set the weights of the links added as desired by the designer.*

This step is a straightforward implementation of the requirement that the home page provide access to each function, and of the requirement that upon completion of each function it should be simple to return to the home page. The procedure is optional because access to and from a home page might be omitted for some MFDs. The designers of the Comanche MFD, for example, have provided buttons on the console that instantly access the home page, rather than providing such access with soft buttons on MFD pages.

Step 7 (optional): Connect sequences

The transformation process as we have described it does not include information to the effect that some functions invariably follow other functions directly, information to the effect that some functions never follow each other, or information to the effect that the probability of one sequence's following another varies with the nature of the two sequences. This information can be added to the graph with the creation of links with stepping stones between the last node of a sequence and the first node of each sequence that follows it. The weight on the link should be proportional to the probability with which the second sequence follows the first. Addition of these links adds the information that navigating through the MFD should make higher-order sequences of functions easy to execute.

This completes our discussion of the novel technique that was developed for transforming an MFD organization problem into a graph that represents the design criteria in a way that is quantitative and precise. Next we describe how such graphs can be used to find good organizations of the MFD pages.

3. Using Graph Partitioning to Solve the Page Organization Problem

The structure produced by the execution of the transformation procedure described above is called by mathematicians a *weighted, directed graph*. Such structures have a number of interesting features, and they have been intensely studied by researchers in a number of disciplines for decades. A surprisingly large number of difficult real-world design problems can be solved fairly well by applying optimization techniques from the fields of mathematics, operations research, and engineering design to graphs of the sort we have derived for the page organization problem. Two of these problems that Dr. Davis has already approached with graph-theoretical tools and genetic algorithms include semiconductor device placement for Texas Instruments, and telecommunication network design for U S West. The results have been highly successful—more successful than competing techniques have been. For these and other reasons, we believe that graph-theoretical approaches to the MFD page organization problem show great promise.

What can one do with the weighted, directed graph produced by the transformation process above? The answer depends on a critical insight. The nodes of the graph represent display items that are visited by a pilot executing mission functions. The links of the graph represent the effort the pilot will expend in navigating the MFD *if the display items at each end of a link are on separate pages*. If we interpret the weight on a link as a load that is imposed in changing pages, then the MFD page organization may be stated simply. It is to group nodes of the graph into pages *so that the total weight of the links that go between pages is minimized*. In graph-theoretical terminology, the problem is to partition the graph into sets of nodes so that the weight of the cut set (the set of links connecting nodes in different sets) is minimized. In graph theory, this problem is termed the problem of graph partitioning.

Graph partitioning is a well-known problem in graph theory. It has myriad real-world applications, and it has been studied for some time. The problem is known to be NP-complete, which means for our purposes that it cannot be solved in realistic amounts of time for problems of any size. It also means that simple, directed techniques for solving the problem are not guaranteed to produce the optimal solution.

Let us illustrate the reduction of the page organization problem to a graph partitioning problem by looking once again at our example. One way of partitioning the graph shown on page D-6 is shown on page D-7. There we see that the display items have been assigned to four pages, satisfying the area constraint that the total area of the items on a page should not exceed 7. There are several points to note about the way we compute the area of a page. First, note that the upper left-hand page has an area of 5, which is the sum of 4 (the area of node B) plus 1 (the area taken up by a button allowing the transition to the upper right-hand page). The stepping-stone node leading to B and the link from it to B lie on B's page, and so they add no area to the total. This is an important point, and must be appreciated in order to understand the graph-theoretical equivalent of the page organization problem. Links and stepping-stone nodes that lie between items on the same page consume no area and incur no pilot access cost. This is because the pilot needn't carry out additional effort to move between display items that are located on the same MFD page. The only pilot cost incurred is represented by links between pages. In the example, there are four such links with a total weight of 33. In the terminology of graph partitioning researchers, these links are the *cut set* for the partitioning, and the design shown on page D-7 has a *cut set cost* of 33. In our terminology, the design shown on page D-7 will cause our pilot to carry out extra work to navigate the MFD, and we can assign a value of 33 to that work. If there is a way to organize the MFD pages so that the sum of the weights on the links is less than or greater than 33, then it is plausible to say that this new way has produced a better or worse design. The lower the cut set weight, the better the design.

It is not difficult to do worse than the design shown on page D-7. Page D-8 shows a design that contains an extra page. This design incurs a cut set cost of 44, and is clearly inferior to the design shown on page D-7. A pilot will expend a greater amount of effort, weighted by criticality and frequency, using an MFD organized as shown on page D-8 than the pilot would expend using an MFD organized as shown on page D-7.

It is also possible to do better than the design shown on page D-7, which was produced by a human who has spent months working with similar problems. The human believed that the design on page D-7 was optimal, principally because of a wealth of experience supporting the rule of thumb that the fewer partitions in a graph partitioning solution, the better the solution is likely to be. However, when our example problem was given to the KL algorithm described below, a solution with cut set weight of 27 was produced. This design is substantially better than the plausible, human-produced design. This design also stands as an exception to the generally useful rule that minimizing the number of partitions implies minimizing the weight of the cut set.

The performance of our human expert here is highly characteristic of human performance on NP-hard design problems. We humans have evolved to perform a wide variety of real-world tasks effectively. We can navigate by means of vision; we can utter and understand highly complex locutions; we can build impressive edifices. It is only recently in our evolutionary process, however, that we have been asked to solve NP-hard problems of the sort being considered here, and our dismal performance in solving those problems is consistent with our historical lack of evolutionary pressure to solve them well. In each of the real-world NP-hard graph-theoretical design problems we have worked with (semiconductor design, telecommunication network design, and graphical layout) we observe again and again that the best results of the most experienced experts are substantially inferior to the global optimum, and to the quality of solution that a reasonable computerized optimization algorithm can achieve. This observation has certainly been borne out by the current example, in which the human, with much effort, generated a solution to a problem with only 15 nodes and 13 links that was 22% worse than the computer-generated solution! When the problem has hundreds or thousands of nodes and there are hundreds or thousands of links, we often find in empirical tests that a human expert's performance lies in the range from 10%-20% below optimal, although these results are strongly problem-dependent.

The difficulty with solving NP-hard problems is not only that they are complex. It is also the case that NP-hard problems are resistant to principled techniques of the sort we humans often employ for problem-solving. There are no expert systems for solving general NP-hard problems quickly, and it is provable that such systems cannot exist. In the face of such essential intractability, researchers have created computerized techniques that have the ability to explore widely the range of solutions to NP-hard problems, using the speed of the computer to consider and evaluate a wider range of solutions than humans can consider. (Genetic algorithms are such techniques.)

The best such algorithms may also contain heuristics that speed up the process or improve on the final results. Such algorithms are being developed and studied at present for a range of NP-hard problems, including the graph partitioning problem. One such algorithm is a variant of the Kernighan-Lin Algorithm (henceforth, "KL"), developed by Kernighan and Lin in the 1970s. The KL algorithm approaches the graph partitioning problem by beginning with a solution in which most of the nodes are on separate pages, then carrying out amalgamations of nodes onto single pages, and swaps of nodes between pages, until a predetermined amount of time has passed. The algorithm in its computerized form explores a great many more solutions than humans would ever be able to explore, and finds reasonably good solutions. KL and its modern variant in which some preprocessing is done in order to pre-establish the KL analog of conceptual clusters when setting up the initial state have been the dominating algorithms in the graph partitioning field.

We used the modern version of KL as a benchmark algorithm in carrying out our work on this phase of the project. The algorithm begins by randomly assigning the graph nodes into partitions. It then tries to reduce the number of pages that contain display items whose total area that exceeds the page size limit. Once it finds a configuration that respects the area constraint, the optimizer enters the reducing total cut set weight phase, making random node exchanges and rejecting any that do not improve the weight of the cut set or that violate the page size limit. The process continues until no pairwise movement of a node or change in the page assignment of a single node can reduce the cut set weight. Typically, one runs this algorithm multiple times and preserves the best solution found among the many runs. The performance of this algorithm is better than most algorithms that have been proposed in the past 20 years, given equal amounts of CPU time. Its performance is highly sensitive to the characteristics of the graph, however. For details of its nature and sensitivities, the reader is referred to the Technical Report in Appendix G, which describes KL and its relative performance more thoroughly and quantitatively than is appropriate here.

For an indication of the algorithm's place in the Display Optimizer architecture, the reader is referred to Figure 1 below. The reader should note too that the KL algorithm and the algorithm in Appendix G have been generalized from their two-partition case to include cases with arbitrary numbers of partitions.

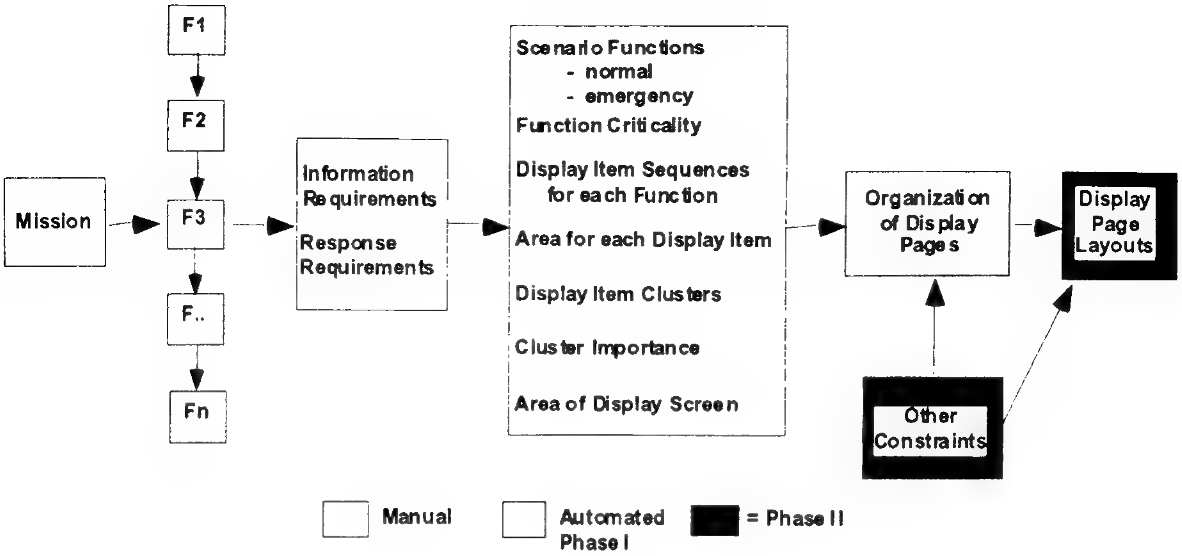


Figure 1. The Display Optimization Procedure

Drs. Shieber, Davis, and Marks have studied the optimization of NP-hard problems for many years, and have, in other domains, compared the performance of algorithms like KL with competing algorithms, including genetic algorithms. It is often the case that significant improvements can be obtained on NP-hard problems when genetic algorithms are created, especially genetic algorithms that exploit the features of their population of solutions, while using search techniques similar to those used by KL to produce new solutions from old ones. Our work on this phase of the project produced a genetic algorithm that appears superior to KL on graphs of the type that are produced by the transformation process described above. We have noted that KL is sensitive to graph type. In Appendix G, Drs. Shieber and Marks present data that our weighted, directed graphs with large numbers of links to and from many of the nodes are not the sort of graphs that KL does well on. And in fact, most of the graph partitioning algorithms extant today have been developed laboriously for graphs that are like those in telecommunications networks and semiconductor devices: graphs with small numbers of links per node or small numbers of partitions. Appendix G describes a genetic algorithm that was designed for another purpose before this contract, and was tailored to this problem by Dr. Marks. Its conversion to the MFD page organization domain was completed near the end of the contract period, and so the algorithm has not been integrated into the software in the prototype display optimizer that was demonstrated at NASA Ames. We tested the algorithm in standalone fashion on page organization problems and other graph partitioning problems, and its solution is superior to its competitors for graphs of the kind we produce for the page organization problem. For a precise description of the algorithm and data on its performance, the reader is referred to Appendix G.

B. Application of the Display Optimizer to the Comanche Test Scenario

In this part of the report, we describe the way information passes from the designer to the optimization system, and we describe the results when our sample scenarios are processed with the Display Optimizer.

1. Flow of Information for Display Optimization

Figure 1 shows the flow of information in the prototype system. The system is intended to assist with the cockpit design process as it is currently carried out. In particular, an integral part of the design of a cockpit is the performance of a task analysis by human factors practitioners. To develop a task analysis, one or more representative mission scenarios are specified which represent typical modes of operation of the helicopter. The human factors analysts break those scenarios down into a hierarchical representation in which phases of the mission are specified and each phase is broken down into segments. Within each segment, the human factors experts specify functions that must be performed to complete that segment. At this stage of the analysis, the specified functions are allocated to, for example, the pilot or the copilot, or to a piece of automated equipment. Each function represents some activity that must be accomplished to complete a particular segment of a phase. The level of detail of a function is that level at which an activity can be specified without reference to specific cockpit design elements and is at an appropriate level of detail for allocation of the function to a particular doer. At this stage of the task analysis, the output is a list of functions. (In Figure 1, these are represented by the boxes labeled F1, F2, etc.) Then, as the design of the helicopter proceeds, each function is analyzed in further detail down to specific detailed actions such as button presses. This analysis is dependent upon a preliminary design of the cockpit.

The Display Optimizer is based on the fact that, when the function analysis is about to be performed, some of the details of other cockpit equipment will be known sufficiently that the designer can specify what information will be required to perform a given function and what responses, or commands to the system, will be required. That is, the designer will be able to associate with each function a sequence of items of information or display elements, which we have called *display items*, required to perform the function. Once the specification of the sequence of display items is complete for each function, the designer can assemble the input data for the Display Optimizer. The input data is summarized in the diagram and consists of the following:

- A list of functions comprising the mission scenario of interest, with emergency procedures flagged
- The criticality of each function

- The sequence of display items required to perform the function
- The screen area required for each display item
- Clusters of display items related conceptually or otherwise
- For each cluster, the importance of keeping the display items in close proximity
- The total area available on the display screen.

These input data are entered by the designer/user into the prototype Display Optimizer through windows and dialog boxes described later. Another parameter used by the Display Optimizer is the frequency of occurrence of each function throughout the scenario. Since this parameter is computed by our prototype Display Optimizer, it need not be entered by the user, although it is included in the input file for the optimization algorithm.

Having carried out the process of specifying the sequences of information accesses and the size of the display items, the designer can then run the Display Optimizer to determine the appropriate organization of information displayed on pages on the MFD computer screen. The output of the Display Optimizer is an allocation of display items to pages that minimizes the number of pages that must be traversed to obtain the information as specified in the display item sequences, while attempting to keep the display items of conceptual clusters in close proximity. It should be noted that a given display item appears on only one page. There are no duplicate presentations.

The Display Optimizer presents its output in several ways. In one presentation mode, the output is a listing of each MFD page, the display items assigned to it, and its links to other pages. An analysis of button presses required throughout the mission scenario is also performed, allowing for direct comparisons of the access effort required by the operator with differing parameter settings, such as size of display screen, or area required by display items, or differing clustering requirements. In another presentation mode, the prototype Display Optimizer presents a crude simulation of the display screen, and listing the display items appearing on each page with buttons with which the user can access pages via links created by the optimization algorithm. Thus, given an MFD page design, the designer/user may use the Display Optimizer to simulate the operator's page access activity throughout the mission scenario. It must be remembered, however, that the current Phase I prototype Display Optimizer only allocates display items to pages; it does not perform any page layout functions. That is one of the major enhancements proposed for our Phase II effort. If the Display Optimizer were integrated into MIDAS, the simulation could be performed by the MIDAS simulation system.

In Appendix F we describe the structure and performance of the prototype Display Optimizer system.

2. Analysis of the Performance of the Algorithm on the Comanche Test Scenario

A listing of functions and the sequences of display items required for each function in our scenario, including the emergency procedures, is shown on pages C-23 through C-28 in Appendix C. The display items and the area assigned to each are listed on page C-29 and C-30. Since the Comanche one hour scenario example problem was run before the PC-based Display Optimizer was completed, an input file called `script` was created manually instead of through the PC user interface. This file was used as input to the version of the algorithm running at Harvard on a Hewlett-Packard workstation. The resulting page organization is shown on pages E-1 through E-4 in Appendix E. It can be seen that this was a very large problem, resulting in 20 pages of display items. Analysis of these results consisted of stepping through each function display item by display item and recording the page to which that display item was assigned. Then we made a tally of button presses required to complete each function. The results of this analysis are shown on pages E-5 through E-11 in Appendix E.

The number of display items per function ranged from one to fifty-seven. We would expect to find that the larger the number of display items accessed, the greater the number of button presses required. One measure of the effectiveness of the optimization might be to compare the number of button presses required to complete each function with a notional worst case in which one button press is required to access each display item in sequence. Figure 2 shows a plot of the number of button presses required to complete each function *vs.* the number of display items required to be accessed for each function. Clearly, the page organization produced by the optimizer is far better than the worst case of one button press per display item. But how much better?

Looking only at the number of display items is not sufficient to allow one to evaluate how well the algorithm does. An important factor to consider in evaluating the results of the optimization is the area required by the display items that must be accessed to complete each function. Display items vary widely (from 2 to 60) in area required. In Figure 3, the number of button presses per function is plotted against the total area required by the display items that must be accessed to complete the function. For example, if a function requires a sequence of display items whose areas are 10, 50, 50, 10, 10, and the number of button presses for that function is 1, the point for that function would be at x=130, y=1. On this graph we have also plotted a reference line which assumes that space is allowed for ten access buttons (with area equal to 3) and that the remaining space on the page is maximally packed. (This assumption does not take the actual distribution of areas into account and so provides a largely unachievable packing density.) This graph provides strong evidence that the algorithm is doing a very effective

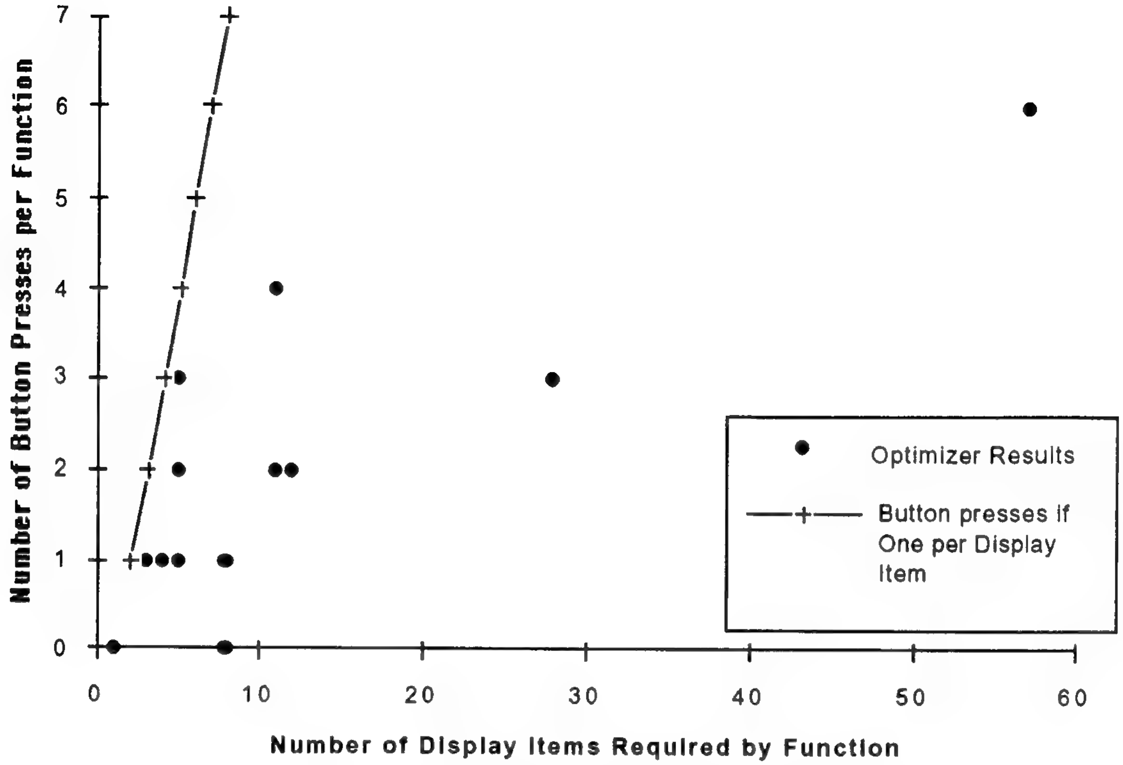


Figure 2. Plot of Button Presses *vs.* Number of Display Items Required for Comanche Example

job of minimizing the number of access button presses required to perform each function. Several of the points lie very close to the line representing something like a theoretical minimum number of button presses. (The reference line is not really the theoretical minimum since we allowed space for 10 access button. However, the reference line may be lower than the true optimum because the widely varying sizes of display items would prevent absolutely complete utilization of the available space.)

We should like to note here that one of the goals discussed at the project kickoff meeting, a quantitative comparison of the existing Comanche MFD design with the design produced by our system, would not be meaningful at this point in the project. The reason for this is that such a comparison will be relevant only if the entire mission scenario is used as input to the design process. It is a trivial matter for our system dramatically to improve on the Comanche design *for the one-hour scenario that was our object of study* because the Comanche helicopter uses two MFD screens, as well as several additional display screens, and each MFD screen has hard buttons above and below it allowing dedicated access to other pages. Because substantially fewer pages are required to execute our one-hour scenario than are required to execute a full Comanche scenario, our

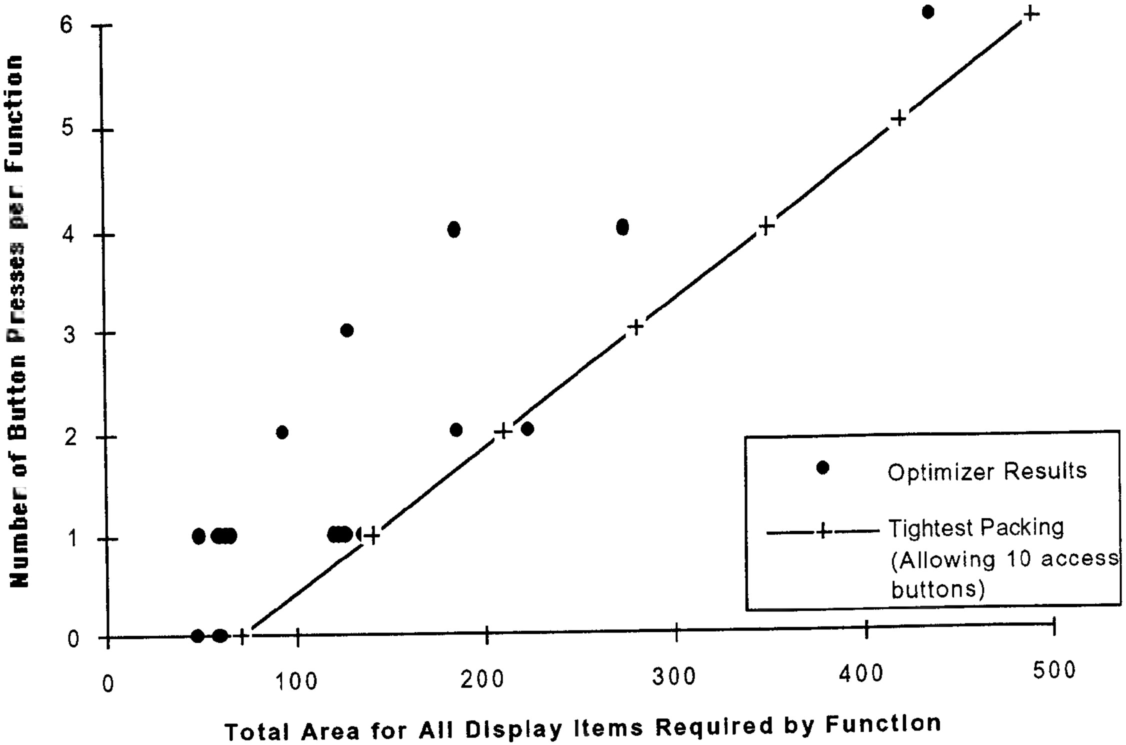


Figure 3. Plot of Button Presses *vs.* Total Area Required for Display Items for Comanche Example

system can use some of the hard buttons to achieve access to the pages that are heavily used in the one-hour scenario, thereby achieving dramatic improvements on the existing Comanche design. This design is not a realistic one, however, because those buttons will be used in the full scenario to provide access to emergency functions and critical functions that our scenario does not include. On the other hand, if we exclude hard buttons from our design, then the Comanche design will be dramatically better than ours, since it has resources for moving directly from any point in the MFD structure to any other point. The home page button is an example of this capability. If our system is required to implement a home page in the way we have described above, by providing soft button access from the end of each sequence to the home page, then the system is working with pages that are effectively smaller in size than the pages that the existing Comanche MFD uses.

A similar problem obtains when one considers the full range of screens used by the Comanche designers. Our research has centered on the design of an MFD system using a single screen. In the Comanche design, two full-sized MFDs are used, with essentially three additional smaller screens. These screens provide capabilities for continuous monitoring, but if they are made available to our system for design based on our one-hour scenario, our system will place display devices on the smaller screens that would not be assigned to those screens when the full scenario is considered. Again, the fact that we are using a subset of the full scenario allows the Display Optimizer unrealistically to show dramatic improvements over the existing Comanche design. On the other hand, if we restrict our system to a single screen, as we have done in our research, then a sequence of display items where the items are large, such as a map and a threat symbology display, would require paging back and forth from one page to another, while in the Comanche as designed those large display items would be displayed simultaneously on two screens. In this case, the Display Optimizer's design would appear unrealistically poor in comparison with the existing Comanche design.

For these reasons and others, we believe that quantitative comparisons of the existing Comanche design and the design our system has produced are not meaningful at this time. To show that the Display Optimizer was capable of handling realistic scenarios, we generated our test problem based on the Comanche task analysis. Meaningful comparisons of our system's designs with the existing Comanche MFD design will not be possible until the entire task analysis has been given to our system. Doing this lies far beyond the scope of the current project resources. (As noted below, we did, however, apply the Display Optimizer to the problem of designing its own interface, and the resulting design was better than the one we had produced!)

Accordingly, in this phase of the project we have devoted our resources to producing the quantitative results described in Appendix G showing that our genetic algorithm produces results better than any competing algorithm for problems like the problem of MFD page organization, and to implementing and analyzing the test case described next.

C. Application of the Display Optimizer to the Display Optimizer User Interface

The Display Optimizer itself has many advantages as a test case. It is close at hand, it uses a single screen as an interface, and it is also a much smaller problem than is the full Comanche task description. For this application, we created a list of functions from the Display Optimizer interface and placed it in the `all_fns.txt` file. We created a list of display items from the interface and placed it in the `all_dis.txt` file. These files are shown on page E-12 in Appendix E. For simplicity, we restricted the functions for this demonstration to those involved with entering the data on which an optimization is to be based. We then set up the sequences of display items required for each function, entered the total page size and areas for each display item, specified clusters of display items that are related conceptually or operationally, and ran the optimizer.

The resulting page organization is shown on page E-13 in Appendix E. One thing to note in these results is that the optimizer used only two pages instead of the three pages that

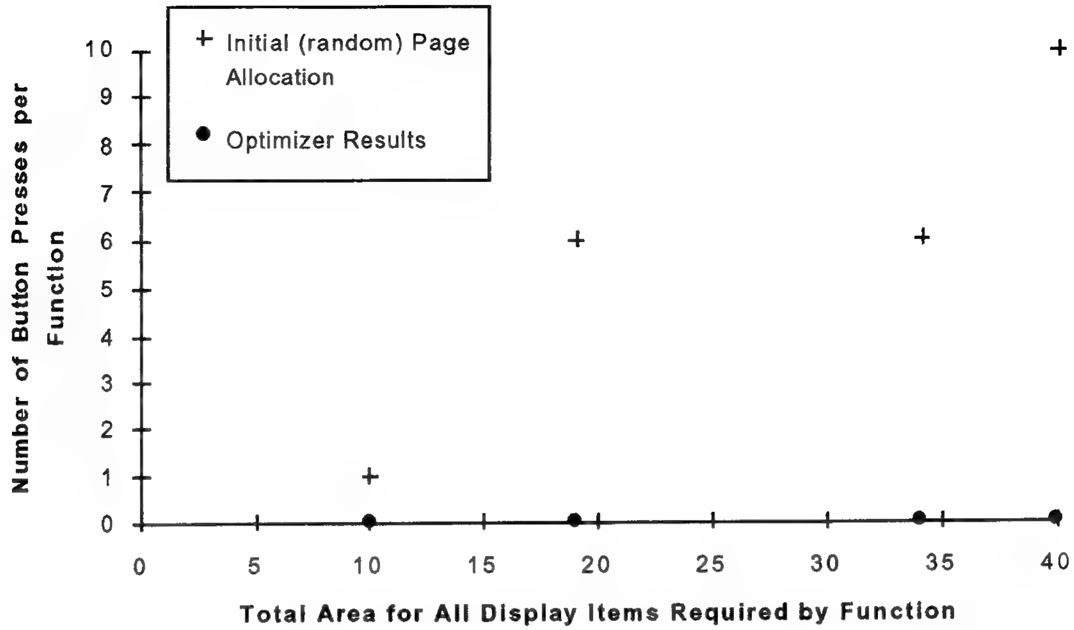


Figure 4. Button Presses *vs.* Total Area Required for Display Items for Display Optimizer Example

we used in our manually created page organization for this application. The interesting thing about this is that in our earlier versions of the Display Optimizer, we had organized the display items just as the Display Optimizer has done. Then, because we needed to present a demonstration that was easy to understand, we wished to demonstrate the basic functions first and then later address the more technical aspects of parameter setting. So, motivated by the need to hide the more technical aspects of the interface, we reorganized the screen, placing the parameter setting display items on a separate page. The result was that by simplifying the appearance of the main display screen allowing for a demonstration that moved more smoothly and didn't get bogged down in technical details before the audience was ready to understand them, we increased the display item access cost by adding a page that required a button press to access and another button press to return.

To get a measure of the effectiveness of the optimization, we looked at the initial random allocation of display items to pages and tallied the number of button presses that would have been required for each function, plotting the result against the total area required by the display items needed for each function. Figure 4 shows the resulting graph. Also plotted are the button presses for each function after optimization. Note that for all functions, no button presses are required to complete a single function. The results of optimization are a clear improvement over the random initial allocation. Although there are no button presses required to complete each function, two button presses are required to complete the entire scenario, as illustrated in the Button Press Analysis on page E-14 in Appendix E. (The initial random page allocation would have required 31 button presses.)

VI. Summary of Results

Our work on Phase I of this project has produced a number of encouraging and interesting results, which we summarize here.

1. Creation of a new procedure for evaluating MFD designs

We believe that the transformation procedure developed by the Harvard/MERL team may have profound implications for the assessment of MFD designs. Prior work on the evaluation of MFD design has been carried out with strong attention paid to human factors features of the design. Some of the constraints mentioned above arose from the valuable work of human factors researchers. If the transformation procedure we have designed preserves the important features of the problem, then we have produced a new tool that provides additional, quantitative procedures for evaluating MFD designs. Put simply, if the transform of one design has a lower cut set weight than the transform of a second, then the first will be easier for the pilot to use, from the point of view of navigating through the MFD pages. The cut set weight measure has to do with the efficiency of the MFD organization, and is a numerical quantity that is easily understood. There are a great many considerations and constraints that must be taken into account when using this measure in order to make it meaningful. We discuss some of them in the Future Work section of this report. Nonetheless, the transformation of the problem into the graph-theoretical domain, a domain that is well-studied and that has provided solutions for a wide variety of other real-world design problems, seems to us to be a major positive outcome of the present work.

2. Production of a working prototype

We created a working software prototype of the Display Optimizer and presented it to the COTR and other interested persons at the Project Review Meeting at NASA Ames in November. This prototype allowed interested persons to see in a hands-on way what is involved in describing and weighting sequences of display item accesses, and in creating the database that the optimization algorithm will use to determine its result. In our view, the prototype provided empirical proof that the techniques we have developed can be made available to cockpit designers without undue difficulty. Indeed, one of our recommendations for future work is that an interface be created so that the Display Optimizer can be linked directly to a computerized version of a mission task analysis, such as the Army's TAWL/TOSS, so that the tedium of entering hundreds of function sequences can be avoided.

3. Development of a better algorithm for solving the transformed version of the problem

In Appendix G we detail a genetic algorithm that outperforms the KL algorithm on graphs that are like the graphs produced by our transform of the page organization problem. This genetic algorithm is sophisticated and uses state-of-the-art techniques, combining the best of genetic algorithm practice with the best of the KL methodology. The result, a hybrid of the genetic algorithm and KL approaches, displays "hybrid vigor", in that its performance is better than that of either of its parents on our test problems.

We wish to note that the algorithm specified in Appendix G currently lacks one feature of a traditional genetic algorithm. While the algorithm manipulates a population of solutions and uses mutation-like operators, it does not yet incorporate a crossover operator. Technically, in the terminology of the field, this makes the algorithm in its current state an evolutionary algorithm lacking one component in order to be classified a genetic algorithm. The reason the specification of our algorithm does not at present include a crossover operator is that the crossover operators we have developed thus far do not improve on the algorithm's results, when results are compared based on equal amounts of CPU processing time. This is a phenomenon we have noted in many other domains. When applying a genetic algorithm to a new domain, some work is required in order to design mutation and crossover operators that exploit the structure of the new domain. Work on crossover operators of this type is at the top of the list for improvements to the genetic algorithm, and we expect that such operators will be discovered with little additional effort. This has certainly been the case in the other problems we have solved with genetic algorithms.

4. Creation of an exploration tool for designers

Another important outcome of our work has been the creation of a system that may be used by designers to test the impact of different design decisions on the organization of the MFD pages. A designer might wish to restrict the depth of the menu structure to three levels. The designer could run the optimization algorithm with this constraint imposed and compare the quality of the result obtained with that obtained when the constraint is eliminated. Similarly, the designer could increase or decrease the size of a display item and study the effect of such decisions on the MFD page organization. The system we have produced facilitates such exploration, together with some assurance that the design obtained by the optimization algorithms we have provided is a good one, given the constraints that are in effect. It takes a human designer a very long time to produce reasonable designs for complicated scenarios, and this is a barrier to contemplation of alternate design decisions. Our system makes the testing and analysis of varying design decisions much simpler to carry out. Since our optimization algorithms may take several hours to run on a problem of moderate size, the Display Optimizer's response will not appear in real time. But the system will give the designer a reasonable answer to a number of "what-if" questions in a few days that have great bearing on the safety of the pilots and the success of the mission.

5. Creation of a general screen-based interface design tool

The techniques we have developed in carrying out the Phase I research are general in nature, and may be applied to other screen-based interfaces. For example, Dr. Marks, who serves on a select National Science Foundation committee on technologies that will be required for effective use of the Information Superhighway, informs us that his committee is recommending strongly that the NSF fund research into menu organization and menu structure. The task of a person accessing information from remote sites on the Information Superhighway is similar in many respects to that of a pilot accessing information through an MFD. Dr. Marks' committee has found our Phase I results to be exciting and stimulating.

VII. Future Work

In this section we describe several areas for future research that we believe would significantly improve the usefulness of the prototype Display Optimizer. These are: enhancements to the transformation process so that additional real-world constraints can be accommodated; enhancements to the graph partitioning algorithms so that results may be found more efficiently and effectively; development of user interfaces so that designers can use the Display Optimizer in a more natural fashion; and addition of a two-dimensional layout module that would generate actual MFD page layouts from the clusters of display items produced by the Display Optimizer in its current form. We describe each of these topics in detail below.

Enhancements to the transformation process

We incorporated a number of different constraints on MFD layout in the current version of the Display Optimizer, in order to show that those constraints can be accommodated, and in order to determine the impact of those constraints on the run time and effectiveness of our graph partitioning algorithms. Some constraints that designers typically satisfy cannot naturally be accommodated by the Display Optimizer in its current form. Examples of such constraints include: allowing the size of display items to be variable and determined during the optimization process; setting a limit on the depth of the menu tree for the MFD; and limiting the number of cross-connections in the MFD menu structure. Accommodating such constraints is quite possible, but it requires modifications to the Display Optimizer as it stands. These modifications and other related changes required to accommodate the real-world aspects of cockpit design form one important area for future work.

Enhancements to the graph partitioning algorithms

We have described evolutionary algorithms developed by members of our team that exceed the performance of the best published graph partitioning algorithms, both in speed and effectiveness, on graphs like those created by our transformation process. The use of evolutionary algorithms has often been found by us to yield such improvements (partly because we design our evolutionary algorithms so that they exploit the best features of existing algorithms). We expect that additional work on our algorithms will yield additional optimization performance. Such improvements in performance would be of use in cockpit design, as well as the other areas of industrial manufacturing and human-computer interaction in which design problems that can be reduced to graph partitioning problems are solved by graph-theoretical means. We believe that additional work on our algorithms will yield considerable benefits for semiconductor designers and designers of human interfaces for the information superhighway, as well as interfaces to other complicated systems.

Our optimization algorithms will need to be extended, however, to handle several features of the real-world problem. We have noted above that the Comanche helicopter has two full-sized MFD screens and three small screens. Some modifications to our algorithm are required in order to expand it to handle multiple screens with different page sizes. We have noted above that the Comanche helicopter uses hard buttons for dedicated page access. Extensions to our algorithm are required in order to assign pages to hard buttons in an optimal way.

We require similar extensions to the Display Optimizer to accommodate that fact that some items should be visible for long amounts of time. The designers of the Comanche helicopter have satisfied this constraint by adding small screens to the display, and it would be interesting to see what a computerized optimization technique would do when given this additional display area to work with.

Our system currently allows each display item to appear on only one page. Modifications would be required in order for the algorithm to allow multiple appearances of display items in order to reduce access costs for those items.

Development of the user interface

We have produced a basic interface to the Display Optimizer in order to demonstrate the viability of our approach to the problem of MFD page configuration. A good deal more can be done to support cockpit designers in their interaction with the Display Optimizer. Potential enhancements are obvious and numerous.

Addition of a two-dimensional layout module

The Display Optimizer assigns display items to MFD pages in accord with the requirements and specifications provided by a cockpit designer. The designer is not able easily to understand the results because the results do not constitute a complete MFD design. An additional module is required in order to accomplish this. What is required is a layout module that takes information sources as input and positions them on the MFD pages, subject to a variety of constraints on usability, consistency, similarity of approach, and so on. Given such a module, a designer will more easily be able to understand the effect on the design of modifying any of the design parameters. Such changes might include: the criticality of various sequences in the scenario, additions and deletions to the set of constraints on the MFD clustering and on the layout itself, and modifications to the size of the MFD pages and the nature of the technologies to be used.

Layout is another problem that is NP-hard. We believe that a successful module for accomplishing layout will have many of the features that our graph partitioning module has: the ability to incorporate and translate user constraints and requirements; the ability to produce high-quality layouts efficiently and effectively through the use of evolutionary algorithms and classical algorithms; and the ability to show the user graphically the effects on the design of the user's requirements.

Development of a two-dimensional layout module would be of benefit to designers in a wide range of fields. In the past two years, Tica Technologies, Inc. has received three queries about two-dimensional layout with evolutionary algorithms. One was from a designer working with CAD/CAM systems for positioning pieces on sheets of metal so that automobile parts may be cut out of metal with minimal wastage. One was from a designer working with CAD/CAM systems for laying out semiconductor devices on a chip with minimal wasted area. The third was from warehouse personnel seeking to position inventory in a warehouse so that a variety of constraints are satisfied. In each of these cases, what is needed is an interface through which the requirements of the problem can be specified and the nature of the items to be positioned in a two-dimensional space can be described; an algorithm that finds good solutions to the positioning problem in reasonable amounts of time; and an interface for presenting the results to the user in the form of a graphical presentation of the design, or to suggest modifications to the requirements so that the system can produce a different, possibly better design. These features, once developed for the cockpit design problem, would be applicable to many other domains.

VIII. Conclusions

In this report we have described the creation of the Display Optimizer, a software system that organizes MFD pages so that pilots can more easily execute the functions involved in completing their missions. We have shown that the system can be used to facilitate the description of mission requirements and specifications so that the system can produce effective MFD page partitions. We have shown that when the system has transformed the problem into a graph partitioning problem, by using an evolutionary algorithm the system is able to produce results that are better than any published algorithm. We have shown that the approach we have taken can be expanded to incorporate a number of additional types of constraints on designs and mission requirements.

In our view, the Display Optimizer shows great promise. Its potential would be enhanced and its commercial viability would be heightened if it were the subject of a Phase II SBIR project. We would be pleased to furnish a proposal to carry out such a project, if invited.

Acknowledgments

This project has been greatly aided by the direction provided by COTR Barry Smith. Mr. Smith's technical and aesthetic sense led us to use MFD design rather than two-dimensional layout as the test problem, persuaded us to use the Comanche MFD as the test device, and persuaded us to create the prototype software Display Optimizer so that interested persons could get a feel for the sort of interactions required in order to use our system. Each of these choices has turned out in retrospect to be the correct one. He has been a positive and responsive COTR, and we are fortunate to have been working with him.

Other members of NASA Ames staff helped us to achieve the results reported here. David Bushnell and Michael Prevost, members of the MIDAS project staff, have been recommending for years that genetic algorithms be linked with MIDAS for cockpit optimization. Anthony Andre provided us with many useful technical and aesthetic comments. Michael Shafto provided us with useful comments. Adam Pease was of help in many ways, including facilitating our access to the Comanche task analysis documents. Kevin Corker provided us with helpful human factors feedback. Yvonne Spector applied her many skills to assuring that our project meetings went without a hitch. We are grateful to all these persons and others we have neglected to thank for their assistance and support.

Bibliography

Andre, A., C. Wickens, and J. Goldwasser 1990. Compatibility and Consistency in Display-Control Systems: Implications for Decision Aid Design. Report ARL-90-13/NASA A3I-90-2, prepared for NASA Ames Research Center.

Bramlette, M. 1991. Genetic Algorithms in Parametric Design of Aircraft. In Davis 1991.

Bui, T., Heigham, C., Jones, C., and Leighton, T. 1989. Improving the performance of the Kernighan-Lin and simulated annealing graph bisection algorithms. *Proceedings of the 26th ACM/IEEE Design Automation Conference*, pp 775-778.

Christensen, J., J. Marks, and S. Shieber 1993a. Algorithms for Cartographic Label Placement. *Proceedings of the American Congress on Surveying and Mapping 1993*, Vol. 1. New Orleans, Louisiana.

Christensen, J., J. Marks, and S. Shieber 1993b. Placing Text Labels on Maps and Diagrams. *Graphics Gems (IV)*, Paul Heckbert, editor. Academic Press, to appear.

Corker, K. and L. Davis 1988. Object Oriented Simulation for Human Performance Evaluation. *Proceedings of the 23rd Annual Conference on Manual Control*, Cambridge, Ma.

Coombs, S. and L. Davis 1987. Genetic Algorithms and Communication Link Speed Design: Implementation Issues. *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, J. Grefenstette editor. Morgan Kaufmann.

Cox, L. A., L. Davis, and Y. Qiu. 1991. Dynamic Anticipatory Routing in Circuit-Switched Telecommunications Networks. *Handbook of Genetic Algorithms*, L. Davis editor. Van Nostrand Reinhold.

Davis, L. 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.

Davis, L., D. Orvosh, A. Cox, and Y. Qiu 1993. A Genetic Algorithm for Survivable Network Design. *Proceedings of the Fifth International Conference on Genetic Algorithms*, Stephanie Forrest, editor. Morgan Kaufmann.

Davis, L. and S. Coombs 1987. Genetic Algorithms and Communication Link Speed Design: Theoretical Considerations. *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, J. Grefenstette editor. Morgan Kaufmann.

Dengler, E., M. Friedell, and J. Marks 1993. Constraint-Driven Diagram Layout. *Proceedings of the 1993 (IEEE/CS) Symposium on Visual Languages.*

Flightfax 1993. *Flightfax Report of Army Aircraft Accidents*, June 1993 cover article.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, Ma.

Holland, J. 1975. *Adaptation in Natural and Artificial Systems*. Originally published by the University of Michigan Press in 1975. Reissued by MIT Press, Cambridge, Ma., 1992.

Karr, C. 1991. Air-Injected Hydrocyclone Optimization via Genetic Algorithm. In Davis 1991.

Kernighan, B. and Lin, S. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* 49(2):291-307.

Kosak, C., J. Marks, and S. Shieber 1991. A Parallel Genetic Algorithm for Network-Diagram Layout. *Proceedings of the Fourth International Conference on Genetic Algorithms*, R. Belew and L. Booker, eds. Morgan Kaufmann.

Kosak, C., M. Marks, and S. Shieber 1993. Automating the Layout of Network Diagrams with Specified Visual Organization. *IEEE Transactions on Systems, Man, and Cybernetics.*

Marks, J. and S. Shieber 1991. The Computational Complexity of Cartographic Label Placement. Center for Research in Computing Technology, Harvard University, TR-05-91.

Ngo, J. T. and J. Marks 1993a. Spacetime Constraints Revisited. Computer Graphics (*Proceedings of SIGGRAPH 1993*).

Ngo, J. T. and J. Marks 1993b. Physically Realistic Motion Synthesis in Animation. Evolutionary Computation 1:3. MIT Press, Cambridge, Ma.

Orvosh, D. and L. Davis 1993. Shall We Repair? Genetic Algorithms, Combinatorial Optimization, and Feasibility Constraints. *Proceedings of the Fifth International Conference on Genetic Algorithms*, Stephanie Forrest, editor. Morgan Kaufmann.

Smith, D. and L. Davis 1985. Adaptive Design for Synthesis. *Proceedings of the 1985 Texas Instruments Conference on VLSI Design.*

Vincow, M., and C. Wickens 1992. Space and the Proximity-Compatibility Principle: Traditional and Computational Model Approaches to Displayed Information. Report ARL-92-7/NASA A3I-92-2, prepared for NASA Ames Research Center.

Appendix A

About Genetic Algorithms

About Genetic Algorithms

Genetic algorithms, invented by John Holland in the late 1960s, are computer techniques for optimization and machine learning based on some features of the biological theory of evolution. “Classical” genetic algorithms were described in detail in Holland 1975. The first and best textbook on the subject is Goldberg 1989. The reader is also referred to Davis 1991, which contains a primer on techniques for using genetic algorithms to solve real-world problems and twelve application case histories. A brief description of the genetic algorithm follows. For more detailed information, the reader is referred to Goldberg 1989 and Davis 1991.

A genetic algorithm solves problems by "evolving" solutions to them. In this approach, solutions to problems are encoded as *chromosomes* that are subject to analogs of the natural processes of survival of the fittest, mutation, and recombination. To solve a problem with a genetic algorithm, one creates a population of chromosomes encoding solutions to the problem and provides an evaluation function that measures any solution’s worth. The algorithm then repeats the cyclic process shown in Figure 1 until it is halted. The cycle begins with the random selection of parents from the population. Although random, the selection process is biased so that better chromosomes in the population are more likely to be chosen for reproduction than less fit ones. The chromosomes chosen for reproduction are cloned to produce children, and the parents are returned to the population. The children are then subjected to random processes of mutation and recombination (also called crossover). The (possibly) modified children are evaluated by the evaluation function, less fit members of the population are deleted to make room for them, and the children are inserted into the population.

This cycle of parent selection, cloning to produce children, modification of the children, evaluation, deletion, and replacement is repeated until a halting criterion is met, at which point the best individual in the population is taken as the genetic algorithm’s solution to the problem. If the techniques used to encode solutions to the problem on chromosomes are appropriate, and if the evaluation function accurately represents how well any chromosome solves the problem, then an initial population of undistinguished chromosomes can evolve to produce better and better solutions, perhaps resulting in a solution better than any that a human might have found. It is important to note that many techniques have been used to accomplish each of the steps of the cycle shown in Figure 1. Which technique to use for which problem, and which parameter settings to use, are questions that are resolved at present more as art than science.

Important features of the genetic algorithm’s approach to optimization include: *randomness* (successive runs of the algorithm may produce quite different solutions); *global search* (the genetic algorithm explores many different types of solutions in parallel, attempting to combine their best features during the recombination part of the cycle); *coadapted solution generation* (the genetic algorithm tends to find solutions with components that have evolved together, rather than modifying components of the solution individually as some optimization procedures do); and *robustness* (the algorithm has been successfully applied to an astonishingly diverse range of problems).

From the time of their invention by Holland in the 1960's until the early 1980s, genetic algorithms were primarily a topic of academic interest. The first important applications appeared when industrial researchers began to use genetic algorithms for optimization purposes in the

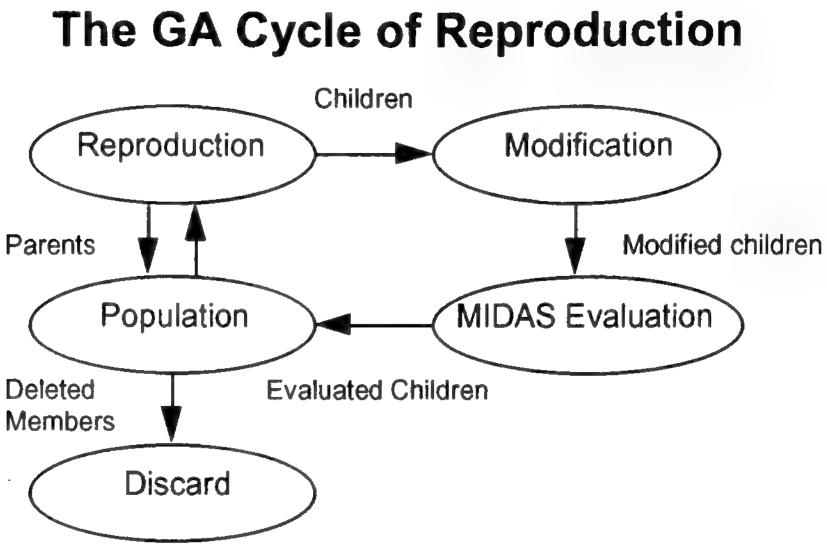


Figure 1. The Genetic Algorithm Cycle of Reproduction with MIDAS as the Evaluator.

early 1980s. At the present time, interest in genetic algorithms for applications is growing rapidly. Approximately one-third of the papers at the 1993 International Conference on Genetic Algorithms described techniques for applying genetic algorithms to real-world problems.

A few of the problems that genetic algorithms are currently solving in the real world include: the *scheduling* of a research facility at Point Mugu Naval Airbase; the *analysis* of mortgage-backed securities for Hyperion Capital Management in New York City; the *discovery of market indicators* for a system at Citibank in London that trades on the foreign currency exchange; and the *design* of fiber optic telecommunications networks for U. S. West, an application that U. S. West estimates will save in the neighborhood of one hundred million dollars by the end of the decade.

Our approach to the current project depends on specializing the classical genetic algorithm, described briefly above, so that it is an effective optimization technique for problems of this type. Specialization is required because the classical genetic algorithm, as described in Holland 1975, includes no knowledge about the problem that the algorithm is solving. Crucial to the success of many real-world applications of genetic algorithms, however, is their incorporation of human heuristics and other knowledge about the problem being solved. In addition, genetic algorithm performance can be greatly enhanced when other optimization techniques are hybridized with the genetic algorithm. The extension of the classical genetic algorithm to accommodate domain knowledge and other algorithms in order to solve real problems well is a central theme of Davis 1991, and is a topic of increasing interest in the field. It was also a central task in this project.

Genetic algorithms have never been applied to cockpit configuration problems, but in the past few years they have developed to a level of sophistication that makes them quite well-suited for such problems. In particular, they have been successfully applied to some simpler, less-constrained layout and configuration problems (many of these applications have been produced by members of our project team). Genetic algorithms have also been successfully used to optimize designs when linked to very large and complicated simulations of performance, as in Bramlette 1991 and Karr 1991. Finally, genetic algorithm practitioners have recently discovered new techniques that greatly improve the performance of genetic algorithms under constraints similar to those in the problem of cockpit configuration (Orvosh and Davis 1993; Davis, Cox, Orvosh, and Qiu 1993).

Given this recent work, and given the increasing sophistication of models of human performance such as MIDAS's representation of pilot performance in a cockpit, it seems highly appropriate to solve problems that are difficult and important by bringing together the genetic algorithm's ability to evolve coadapted solutions to problems with many constraints and highly-developed models of human performance such as MIDAS. In this way, designs will be produced that are tailored to the way pilots use cockpit information displays, as modeled by MIDAS.

Appendix B

The MIDAS Interface File Format

Input/Output File Specification

The Input File

The input file, *scripts*, consists of the declaration of the page area size and the information sources, the scripts, and other grouping constraints. A line that starts with the '#' symbol is treated as a comment

The Area Limit and Information Sources

First, the area of a page is specified. The next line declares the number of information sources, followed by a listing of the information sources, one on each line. Each entry should contain the identification number of the information source and the area that the information source takes. The identification number is expected to be a non-negative integer (if it's more convenient, we can use character string names instead), and the area is expected to be a real number.

The scripts

We have two different types of scripts: regular and emergency. Because one may wish to jump to an emergency script at any time, the beginning of each emergency is implicitly connected to all other information sources (including those in other emergency scripts). We shall first declare all the regular scripts before all the emergency scripts. Each regular script entry begins with a special character 's', followed by the frequency value and the criticality value. Both numbers are positive integers. The following lines contain the information sources needed for the script, one per line. The emergency script entries have the same format as the regular scripts, except that each entry begins with a special character 'e'. As before, the frequency and criticality weights are applied uniformly to the actual scripts; however, when connecting other nodes to the head of the emergency scripts, the program will automatically normalize the frequency and criticality values for these links.

Clustering Constraints

Here, we specify the sets of information sources that ought to go together onto the same page (for conceptual reasons or because they are needed simultaneously). For each cluster, the entry begins with a special character 'c', followed by a positive number representing the relative importance of the cluster. Then, the identification number of each information source is listed one per line.

An Example

Suppose we have 5 information sources: 0,1,2,3,4. We also have two regular scripts and one emergency scripts. Furthermore, information sources 0 and 1 are conceptually related, and 2 and 3 are needed simultaneously.

type	script	frequency	criticality
regular	0 → 2 → 3	20	30
regular	1 → 2 → 4	10	5
emergency	0 → 3 → 5	10	100

Sample File *scripts*

```
# the area of a page
50.00
# number of information sources.
5
# information sources id, followed by the area needed.
0        24.56
1        32.34
2        21.75
3        10.97
4        16.63
# first script: 0 → 2 → 3
s        20        30
0
2
3
# second script: 1 → 2 → 4
s        10        5
1
2
4
# emergency script: 0 → 3 → 5
e        10        100
0
3
5
# clusters:
c        10
0
1
c        15
2
3
```

The Output File

The output file lists the content of each page (i.e. information sources) and the connections of that page to other pages.

An Example

```
Page 1 contains:
     Information sources:
                 0
                 1
     Connects to pages:
                 2
Page 2 contains:
     Information sources:
                 2
                 3
                 4
     No connection to other pages.
```

Appendix C

Comanche Mission Scenario

Scenario Functions for Comanche Mission

Perform Before Takeoff Check
Monitor Threat
Perform Navigation (Contour) (C)
Receive Digital Movement Message
Prepare and Send Digital Movement Report
Receive Digital Message
Select Navigation waypoint
Monitor Threat
Perform Navigation (Contour)(C)
Monitor Threat
Perform Navigation (NOE)
Select Overwatch Position
Monitor Threat
Perform Navigation (NOE) (C)
Set up Automatic Ground Search Configuration
Review Automatic Search Target Frames
Select Navigation Waypoint
Perform Navigation (NOE) (C)
Prepare and Send Digital Movement Report
Monitor Threat
Perform Navigation (NOE) (C)
Select Overwatch Position
Monitor Threat
Perform Navigation (NOE) (C)
Set Up Automatic Ground Search Configuration
Review Automatic Search Target Frames
Set up Automatic Ground Search Configuration
Review Automatic Search Target Frames
Prepare and Send Digital Free Text Message
Select Observation Point
Perform Navigation (NOE) (C)
Prepare and Send Digital SPOT Report (Ground Search)
Monitor Threat
Select Overwatch Position
Receive Digital Message
Prepare and Send Digital SPOT Report (Ground Search)
Receive Digital Message
Prepare and Send Digital Free Text Message
Select Navigation Waypoint
Prepare and Send Digital Movement Report
Monitor Threat
Perform Navigation (NOE) (C)
Select Overwatch Position

Select Observation Point
Perform Navigation (NOE)
Set Up Automatic Ground Search Configuration
Review Automatic Search Target Frames
Prepare and Send Digital Free Text Message
Prepare and Send Digital SPOT Report (Ground Search)
Receive Digital Message
Select Observation Point
Prepare and Send Digital Free Text Message
Prepare and Send Digital Movement Report
Monitor Threat
Perform Navigation (NOE)
Set Up Automatic Ground Search Configuration
Review Automatic Search Target Frames
Select Overwatch Position
Set Up Automatic Ground Search Configuration
Review Automatic Search Target Frames
Select Transmit Radio
Monitor Gun Engagement
Select Transmit Radio
Receive External Voice Communication
Prepare and Send Digital Free Text Message
Prepare and Send Digital Message, BDA Report
Prepare and Send Digital Movement Report
Select Observation Point
Perform Navigation (NOE)
Set Up Automatic Ground Search Configuration
Perform Search, Slew Mode (C)
Review Automatic Search Target Frames
Prepare and Send Digital SPOT Report (Ground Search)
Receive Digital Movement Message
Prepare and Send Digital Movement Report
Select Navigation Waypoint
Prepare and Send Digital Movement Report
Monitor Threat
Perform Navigation (NOE) (C)
Select Overwatch Position
Monitor Threat
Set Up Automatic Ground Search Configuration
Review Automatic Search Target Frames
Select Observation Point
Prepare and Send Digital Free Text Message

Comanche Emergency Procedures Included in Mission Scenario

Respond to Advisory Alert (Stored)
Respond to Caution Alert (Stored)
Respond to Warning: Auto Flight Control System, Nonrecoverable Failure
Respond to Warning: Auto Flight Control System, Recoverable Failure
Respond to Warning: Engine Fire
Respond to Warning: Engine Out, Inflight, Nonrecoverable Failure
Respond to Warning: Engine Out, Inflight, Recoverable Failure
Respond to Warning: Primary Flight Control System, Nonrecoverable Failure
Respond to Warning: SPU Fire
Respond to Warning: Weapons Bay Fire
Review Advisories
Review Cautions

2.2.2.4.1.2 RAH-66 COMANCHE ARMED RECONNAISSANCE MISSION TIMELINE.

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
	AA	Ground	Mission Planning based on factors of METT-T (Mission, Enemy, Troops, Terrain and Weather, and Time), utilizing the Integrated Mission Support Station (IMSS) Mission Planning Function	1.1,1.2
	AA	Ground	Preflight Briefing - Includes Flight Path Location of friendly forces Terrain, Situation, Threat Observation Points (OP's) FARP Locations CEOI (Communications and Electronics Operational Instructions) Team Responsibilities	1.1,1.3,1.4
	AA	Ground	Preflight Inspections	1.5,1.6,1.7,1.8
	AA	Ground	Preflight Activities Start Engines Initialize Avionics systems Enter Aircraft Status data Mission Data Load Review and Verify system status	2.2 2.3 1.9,1.10 1.11
	AA	Ground	Commun Transmit/Receive - ECHO 7 PAPA 03 Establishes Communications with Army Airspace Management element; Reports planned flight route via digital Communications network	2.1
	AA	Ground	Commun Transmit/Receive - PAPA 03 Requests/Receives clearance for Team liftoff from Army Airspace Management via digital Communications	2.4
	AA	Ground	Commun Transmit - PAPA 03 Transmits digital signal to PAPA 05 to coordinate Team liftoff	1.12 2.6
00:00	AA	HIGE	Team liftoff	
00:20	Enroute AA - FAA	HIGE - Low Level	Team Transitions to Low Level Flight	
	Enroute AA - FAA	Low Level 9 km	Execute flight operations - Low Level Flight	3.1

TIME	LOCATION	DISTANCE/ FLIGHT <u>MODE</u>	ACTIVITIES	CROSS REF SEGMENT NUMBER
03:10	Enroute AA - FAA	Low Level - Contour	Team Transitions to Contour Flight	
	Enroute AA - FAA	Contour 7 km	Execute flight operations - Contour Flight	3.1
05:30	Enroute AA - FAA	Contour	Commun Receive - PAPA 03 Receives digital message from Squadron TOC to bypass FAA and receive briefing update while enroute	
05:45	Enroute AA - FAA	Contour	Commun Transmit - PAPA 03 Relays orders to bypass FAA to PAPA 05 via digital Communications network	
	FAA	Contour	PAPA 03, 05 pass over FAA location	
06:40	FAA	Contour	Commun Receive - PAPA 03, 05 Receive briefing update via digital data burst as they pass over FAA location	4.3
	FAA	Contour	PAPA 03, 05 Select NAV waypoint: OP1	9.1
	Enroute FAA - OP1	Contour	PAPA 03, 05 Review mission briefing update as they continue on to OP 1 area	
	Enroute FAA - OP1	Contour	Select NAV waypoint: OP 1	
	Enroute FAA - OP1	Contour 18 km	Execute flight operations - Contour Flight	
14:00	Enroute FAA - OP1	Contour - NOE	PAPA 03, 05 Transition to NOE flight	9.6
	Enroute FAA - OP1	NOE/BO 8 km	Execute flight operations - NOE/BO	
	Enroute FAA - OP1	NOE	PAPA 03, 05 Maneuver NOE to OP 1 area	
23:00	OP 1	NOE	PAPA 03, 05 Arrive OP 1 area	10.1
	OP 1	NOE	PAPA 03, 05 Maneuver NOE to identify and assume individual observation/ covering positions within OP 1 area	
	OP 1	NOE - HIGE	PAPA 03, 05 Transition to masked hover	

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
23:25	OP 1	HOGE	PAPA 03 Unmasks to search for threat activity; PAPA 05 Hovers in nearby position from which covering overwatch and support may be provided and situational awareness may be maintained	11.1
	OP 1	HOGE	PAPA 03 Monitors sensors; Conducts sensor scan	11.1
	OP 1	HOGE	Monitors terrain; Searches for threat activity	
23:34	OP 1	HIGE	PAPA 03 Remasks; Transitions to masked hover	
	OP 1	HIGE	PAPA 03 Executes hover hold; Reviews/Evaluates sensor data - No threat activity detected	↓
26:35	OP 1	HIGE	Commun Transmit - PAPA 03 Coordinates with PAPA 05 via preplanned digital signal to depart OP 1 area and continue recon route towards OP 2	10.1
	OP 1	HIGE	PAPA 03, 05 Select NAV waypoint: OP 2	
26:50	OP 1	HIGE - NOE	PAPA 03, 05 Depart OP 1	10.4,11.3
26:52	OP 1	NOE	Commun Transmit - PAPA 03 Transmits digital signal to Squadron TOC to report Team's departure from OP 1	
	Enroute OP1 - OP2	NOE/BO	Maneuver NOE to OP 2 using bounding overwatch (BO - maneuver under cover and concealment while providing mutual overwatch), performing reconnaissance while enroute by means of external observations, terrain monitoring, and searching for threat activity	
	Enroute OP1 - OP2	NOE/BO 5 km	Execute flight operations - NOE/BO	
	Enroute OP1 - OP2	NOE	Monitor Sensors/Displays/Communications	
33:35	OP 2	NOE	PAPA 03, 05 Arrive OP 2 area	↓
	OP 2	NOE - HIGE	PAPA 03, 05 Maneuver NOE to identify and assume individual observation / covering positions within OP 2 area; Transition to masked hover	↓

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
33:55	OP 2	HOGE	PAPA 03 Unmasks to search for threat activity; PAPA 05 Hovers in nearby position from which covering overwatch and support may be provided and situational awareness may be maintained	12.1
	OP 2	HOGE	PAPA 03 Monitors sensors; Conducts sensor scan; Observes terrain, Searches for threat activity	12.1
34:04	OP 2	HOGE - HIGE	PAPA 03 Remasks Transitions to masked hover	
	OP 2	HIGE	PAPA 03 Executes hover hold; Reviews/Evaluates sensor data - PAPA 03 sensors detect threat activity at a distance of 7-8 km. Unable to determine threat force composition due to distance	
36:35	OP 2	HIGE	Commun Transmit - PAPA 03 Communicates with PAPA 05 concerning observations via digital data burst; PAPA 03 Coordinates with PAPA 05 to maneuver to unmask and observe from nearby location, PAPA 05 to unmask to search for additional threat activity in an area not covered by PAPA 03 scan	10.2
	OP 2	HIGE - NOE	PAPA 03, 05 Maneuver NOE to assume new observation/covering positions within OP 2 area; While maneuvering, PAPA 03 begins formulation of digital SPOT Report	
37:10	OP 2	HIGE	PAPA 03 Hovers in nearby position from which covering overwatch and support may be provided and situational awareness may be maintained. PAPA 05 Unmasks to search for threat activity	10.4
	OP 2		PAPA 05 Monitors sensors; Conducts scan	
37:19	OP 2		PAPA 05 Remasks and evaluates sensor data - Threat detected but cannot be classified; No additional threat forces detected. Long range AD radar detected (SA-15; search mode)	

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
38:50	OP 2	HIGE	Commun Receive - PAPA 03 Receives digital report from PAPA 05 to relay observation data; PAPA 05 unable to classify threat due to distance; No additional threat elements have been detected	
39:20	OP 2	HIGE	Commun Transmit - PAPA 03 Formulates and Transmits intelligence data (SPOT) report to Squadron TOC via digital Communications network to report threat detection, location of threat activity, and direction of threat movement	↓
40:10	OP 2	HIGE	Commun Receive - PAPA 03 Receives digital message from Squadron TOC commanding PAPA 03, 05 to continue planned reconnaissance route to OP 3 in order to gain a closer look at the threat force	10.2
40:15	OP 2	HIGE	Commun Transmit - PAPA 03 Coordinates with PAPA 05 via preplanned digital signal to depart OP 2 area and continue recon route to OP 3	
	OP 2	HIGE	PAPA 03, 05 Select NAV waypoint: OP 3	
40:25	OP 2	HIGE - NOE	PAPA 03, 05 Depart OP 2	10.4
40:27	Enroute OP2 - OP3	NOE	Commun Transmit - PAPA 03 Transmits digital signal to Squadron TOC to report Team departure from OP 2	
	Enroute OP2 - OP3	NOE/BO 6 km	Maneuver NOE to OP 3 using bounding overwatch (BO - maneuver under cover and concealment while providing mutual overwatch), performing reconnaissance while enroute by external observations, terrain monitoring, and searching for threat activity	
	Enroute OP2 - OP3	NOE/BO	Execute flight operations - NOE/BO	
45:20	OP 3	NOE	PAPA 03, 05 Arrive OP 3; Maneuver NOE to identify and assume observation/covering positions within OP 3 area; Transition to masked hover	Ø Ø

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
45:30	OP 3	HOGE	PAPA 03 Unmasks to search for threat activity; PAPA 05 Hovers in nearby position from which covering overwatch and support may be provided and situational awareness may be maintained	12.1
	OP 3	HOGE	PAPA 03 Monitors sensors; Conducts sensor scan	
	OP 3	HOGE	Observes terrain, Searches for threat activity	12.1
45:40	OP 3	HOGE - HIGE	PAPA 03 Remasks; Transitions to masked hover	
	OP 3	HIGE	PAPA 03 Executes hover hold; Reviews/Evaluates sensor data - Threat identified as tanks and armored recon vehicles. Sensors detect the presence of threat 2S6 and SA-15 AD radar in search mode	
49:25	OP 3	HIGE	Commun Transmit - PAPA 03 Transmits observation data to PAPA 05 via digital Communications network	
50:25	OP 3	HIGE	Commun Transmit - PAPA 03 Transmits intelligence data (SPOT) report to Squadron TOC via digital Communications network	↓
51:05	OP 3	HIGE	Commun Receive - PAPA 03 Receives digital message (change of Mission Command) from Squadron TOC commanding team to break preplanned recon route and maneuver closer to the threat forces to observe and monitor, but not to enter into engagement	10.2
	OP 3	HIGE	PAPA 03 Evaluates terrain via Terrain Map Display (TMD) to identify potential location for new observation position (OP 3a)	
51:45	OP 3	HIGE	Commun Transmit - PAPA 03 Transmits digital message to PAPA 05 to identify new observation position (OP 3a) area and to coordinate team's departure from OP 3	
51:55	Enroute OP3 - OP3a	NOE/BO	PAPA 03, 05 Depart OP 3;	10.4,11.3

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
52:00	Enroute OP3 - OP3a	NOE	Commun Transmit - PAPA 03 Transmits digital signal to Squadron TOC to report Team's departure from OP 3	
	Enroute OP3 - OP3a	NOE/BO 2 km	Maneuver NOE to locate OP 3a using bounding overwatch (BO - maneuvering under cover and concealment while providing mutual overwatch), performing reconnaissance while enroute by external observations, terrain monitoring, and searching for threat activity. Flight is conducted head-up/eyes-out once the pre-planned flight path is broken	10.2 10.4,11.3
	Enroute OP3 - OP3a	NOE	Monitor Sensors/Displays/Communications	
54:10	OP 3a	NOE - HIGE	PAPA 03, 05 Arrive OP 3a; Maneuver NOE to identify and assume observation/covering positions within OP 3a area; Transition to masked hover	
54:30	OP 3a	HOGE	PAPA 03 Unmasks to search for threat activity; PAPA 05 Hovers in nearby position from which covering overwatch and support may be provided and situational awareness may be maintained	11.1/11.2
	OP 3a	HOGE	PAPA 03 Initiates sensor scan (stare mode)	
54:35	OP 3a	HOGE	Threat BRDM II armored personnel carrier/reconnaissance vehicle emerges from the tree line of nearby woods, approximately 700m from PAPA 03. BRDM detected by visual sighting of muzzle flash by PAPA 03 copilot as the BRDM fires its 14.5 mm gun at PAPA 03	12.3 Ø Ø
	OP 3a	HOGE - HIGE	PAPA 03 Initiates evasive maneuvers to evade gunfire from BRDM;	
		HOGE - HIGE	PAPA 03 Unstows 20 mm gun from Low observable stowed position while maneuvering to evade BRDM gunfire; PAPA 05 observes PAPA 03 chance engagement with threat BRDM, Initiates input to unstow 20 mm gun	13.7

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
54:38	OP 3a	HOGE - HIGE	Commun Transmit - PAPA 03 Alerts PAPA 05 to detection of BRDM, receipt of fire, and BRDM location via verbal message while maneuvering to remask	
		HOGE - HIGE	PAPA 03 Fires 20 mm gun to suppress BRDM fire while maneuvering to remask and gain concealment; BRDM maneuvers to gain concealment in nearby treeline	13.7
	OP 3a	HOGE - HIGE	PAPA 03 Maneuvers away from BRDM to remask while firing 20 mm gun at target	
	OP 3a		PAPA 05 Maneuvers to fully unmask and engage BRDM II; Engages BRDM II with 20 mm gun	
	OP 3a		BRDM II destroyed by combined 20 mm gunfire from PAPA 03 and 05	
54:48	OP 3a	NOE - HIGE	PAPA 03, 05 Transition to masked hover; Commun Transmit/Receive-PAPA 03 verifies BRDM destruction via verbal message with PAPA 05	
55:10	OP 3a	HIGE	Commun Transmit - PAPA 03 Coordinates with PAPA 05 via preplanned digital signal to maneuver to new position nearby	10.2
	OP 3a	NOE	PAPA 03, 05 Maneuver NOE to identify and assume new observation/covering positions nearby	
55:25	OP 3a	NOE	Commun Transmit - While maneuvering NOE, PAPA 03 reports to Squadron TOC via digital message to report the engagement and destruction of BRDM II vehicle; Reports that PAPA 03, 05 are maneuvering to locate new observation position within OP 3a area to continue their reconnaissance	
56:20	OP 3a	NOE - HIGE	PAPA 03, 05 Maneuver into new observation/covering positions within OP 3a area; Transition to masked hover	

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER
56:30	OP 3a	HOGE	PAPA 03 Unmasks to search for threat activity; PAPA 05 Hovers in nearby position from which covering overwatch and support may be provided and situational awareness may be maintained	12.4
	OP 3a	HOGE	Monitors sensors (stare mode), observes threat vehicles	
56:39	OP 3a	HOGE - HIGE	PAPA 03 Remasks; Transitions to masked hover	12.4
	OP 3a	HIGE	PAPA 03 Executes hover hold; Reviews/Evaluates sensor data	
59:20	OP 3a	HIGE	Commun Transmit - PAPA 03 Formulates and Transmits digital intelligence data report to Squadron TOC	
	OP 3a	HIGE	PAPA 03, 05 Remain in masked hover to await orders from Squadron TOC	
1:00:05	OP 3a	HIGE	Commun Receive - PAPA 03 Receives digital message from Squadron TOC commanding PAPA 03, 05 not to engage threat but to continue planned reconnaissance route to OP 4	10.1
1:00:15	OP 3a	HIGE	Commun Transmit - PAPA 03 Transmits digital signal to PAPA 05 to coordinate the Team's departure from OP 3a area to re-establish the preplanned recon route to OP 4	
	OP 3a	HIGE	Select NAV waypoint: OP 4 Select flight route to re-establish preplanned recon route	
1:00:25	OP 3a	HIGE - NOE	PAPA 03, 05 Depart OP 3a	
1:00:27	Enroute OP3a - OP4	NOE	Commun Transmit - PAPA 03 Transmits digital signal to Squadron TOC to report the Team's departure from OP 3a	10.4,11.3
	Enroute OP3a - OP4	NOE/BO	Maneuver NOE to OP 4 using bounding overwatch (BO - maneuvering under cover and concealment while providing mutual overwatch), performing reconnaissance while enroute by means of external observation, terrain monitoring, and searching for threat activity	

TIME	LOCATION	DISTANCE/ FLIGHT MODE	ACTIVITIES	CROSS REF SEGMENT NUMBER	
	Enroute OP3a - OP4	NOE/BO 6 km	Execute flight operations - NOE/BO		
	Enroute OP3a - OP4	NOE	PAPA 03, 05 re-stow 20 mm gun to low observable stowed position to decrease risk of detection		
	Enroute OP3a - OP4	NOE	Monitor Sensors/Displays/Communications		
1:07:30	OP 4	NOE - HIGE	PAPA 03, 05 Arrive OP 4; Maneuver NOE to identify and assume observation/covering positions within OP 4 area; Transition to masked hover	10.1	10.4,11.3
1:07:50	OP 4	HIGE	Commun Transmit - PAPA 03 Coordinates with PAPA 05 via pre-planned digital signal; Directs PAPA 05 to unmask		
1:08:05	OP 4	HIGE	PAPA 03 Hovers in nearby position from which covering overwatch and support may be provided and situational awareness may be maintained; PAPA 05 Unmasks to search for threat activity	10.4	11.1
	OP 4		PAPA 05 Executes sensor scan; Searches for threat activity		
1:08:16	OP 4	HIGE	PAPA 05 Remasks; Executes hover hold; Evaluate sensor data - Threat activity not detected. Threat long range radar detected (SA-15) in search mode; PAPA 05 Assesses headings of SA-15 radar		
1:11:35	OP 4	HIGE	Commun Receive - PAPA 03 Receives digital data burst from PAPA 05 relaying PAPA 05 observation data		
	OP 4	HIGE	PAPA 03, 05 Remain in masked hover; PAPA 03 Reviews digital message received from PAPA 05		
1:12:10	OP 4	HIGE	Commun Transmit - PAPA 03 Transmits digital signal to PAPA 05 to coordinate Team departure from OP 4 to continue planned recon route to OP 5	10.1	
1:12:20	OP 4	HIGE - NOE	PAPA 03, 05 Select NAV waypoint: OP 5; Depart OP 4		10.4,11.3
1:12:23	Enroute OP4 - OP5	NOE	Commun Transmit - PAPA 03 Transmits digital signal to Squadron TOC to report Team's departure from OP 4		

RA EGI 'SUMMARY

PHASE 10: Movements in the Reconnaissance/Battle Area

SEGMENT 4: Perform Bounding Overwatch in Reconnaissance/Battle Area

PILOT			COPILOT		
DISCRETE FIXED	DISCRETE RANDOM	CONTINUOUS	DISCRETE FIXED	DISCRETE RANDOM	CONTINUOUS
		Fly NOE		Monitor Threat (TSD)	Perform Navigation (NOE) (TSD)
		Perform Navigation (NOE) (HMD)			
				Initiate Cockpit Communication (Copilot)	
			Select Overwatch Position (TSD)		
Transition to Hover					
Establish Overwatch Position					
		Hover Masked			
		Monitor Bounding Aircraft (Pilot)			Monitor Bounding Aircraft (Copilot)
	Initiate Cockpit Communication (Pilot)				

RAI :EGA SUMMARY

PHASE 12: Acquire Ground Targets in Reconnaissance/Battle Area

SEGMENT 1: Acquire Ground Target, Automatic Search From Observation Point/Battle Position

PILOT			COPILOT		
DISCRETE FIXED	DISCRETE RANDOM	CONTINUOUS	DISCRETE FIXED	DISCRETE RANDOM	CONTINUOUS
		Hover Masked			
			Set Up Automatic Ground Search Configuration		
	Initiate Cockpit Communication (Pilot)				
				Initiate Cockpit Communication (Copilot)	
Unmask Aircraft (Search)					
		Hover Unmasked			
Mask Aircraft					
Transition to NOE					
		Fly NOE	Review Automatic Search Target Frames		
				Initiate Cockpit Communication (Copilot)	
			Prepare and Send Digital SPOT Report (At Conclusion of Ground Search)		

Mission Scenario Timeline

RAH-66 Comanche Armed Reconnaissance Mission Timeline (pg 2-27)

time	Scenario summary	Pilot tasks	Copilot tasks	Segment
	Preflight Activities		Perform External Voice Communication	2.1
			Receive External Voice Communication	
		Start Left Engine		2.2
		Start Right Engine		
		Monitor Aircraft Health (C)		
		Check/Manage NVPS		2.3
		Check/Manage TAS		
		Check/Manage ASE		
			Perform External Voice Communication	2.4
			Receive External Voice Communication	
		Perform Before Takeoff Check (Pilot)	Perform Before Takeoff Check (Copilot)	2.5
0:00:00	Liftoff	Perform Hover		
		Hover Masked (C)		
		Taxi Aircraft (C)		
		Rendezvous with Other Company Aircraft		
0:03:10	Execute flight operations - Contour	Transition to Contour	Perform External Voice Communication	2.6
		Establish Level of Flight		
		Fly Contour (C)		
		Monitor Threat (Unaided)	Monitor Threat (TSD)	3.1
		Fly Contour (C)	Perform Navigation (Contour) (TSD) (C)	
0:05:30		Perform Navigation (Contour) (HMD) (C)	Receive Digital Movement Message	
0:05:45			Prepare and Send Digital Movement Report	
0:06:40	Select Waypoint OP1	Receive and Review Flight Plans/Overlays	Receive Digital Message	4.3
			Select Navigation waypoint (TSD)	10.1
		Monitor Threat (Unaided)	Monitor Threat (TSD)	3.1
		Fly Contour (C)	Perform Navigation (Contour) (TSD) (C)	
		Perform Navigation (Contour) (HMD) (C)		
0:14:00	Execute flight operations-NOE/BO	Transition to NOE		
		Monitor Threat (Unaided)	Monitor Threat (TSD)	
		Fly NOE (C)		10.4
		Perform Navigation (NOE) (HMD) (C)	Perform Navigation (NOE) (TSD)	
		Transition to Hover	Select Overwatch Position (TSD)	
		Establish Overwatch Position		
		Hover Masked (C)		
0:23:00	Arrive OP1 area	Transition to NOE		
		Monitor Threat (Unaided)	Monitor Threat (TSD)	
		Fly NOE (C)		
		Perform Navigation (NOE) (HMD) (C)	Perform Navigation (NOE) (TSD) (C)	

Mission Scenario Timeline

time	Scenario summary	Pilot tasks	Copilot tasks	Segment
		Transition to Hover		
		Hover Masked (C)		
0:23:25	Unmask to search for threat	Unmask Aircraft (Search)	Set up Automatic Ground Search Configuration	11.1
		Hover Unmasked (C)		
		Perform search (Unaided)	Review Automatic Search Target Frames	
0:23:34	Remasks; no threat detected	Mask Aircraft		
		Hover Masked (C)		
0:26:35	Select Waypoint OP2		Select Navigation Waypoint (TSD)	10.1
0:26:50	Depart OP1		Perform Navigation (NOE) (TSD) (C)	
0:26:52	Comm Transmit		Prepare and Send Digital Movement Report	
	Maneuver NOE/BO to OP2	Transition to NOE		
		Monitor Threat (Unaided)	Monitor Threat (TSD)	
		Fly NOE (C)		10.4
		Perform Navigation (NOE) (HMD) (C)	Perform Navigation (NOE) (TSD)	
		Transition to Hover	Select Overwatch Position (TSD)	
		Establish Overwatch Position		
		Hover Masked (C)		
		Monitor Threat	Monitor Threat (TSD)	11.3
		Transistion to NOE		
		Fly NOE (C)		
		Perform Navigation (NOE) (HMD) (C)	Perform Navigation (NOE) (TSD) (C)	
			Set Up Automatic Ground Search Configuration	
0:33:35	Arrive OP2		Review Automatic Search Target Frames	
	Transition to Masked Hover	Transition to Hover		
		Mask Aircraft		
		Hover Masked (C)	Set up Automatic Ground Search Configuration	12.1
0:33:55	Search for threat	Unmask Aircraft (Search)		
		Hover Unmasked		
		Mask Aircraft		
0:34:04	Detect threat at 7-8 km distance	Hover Masked (C)	Review Automatic Search Target Frames	
0:36:35	Comm Transmit/Maneuver NOE	Transition to NOE	Prepare and Send Digital Free Text Message (?)	
	Maneuver to new position		Select Observation Point (TSD)	10.2
		Fly NOE (C)		
		Perform Navigation (NOE) (HMD)	Perform Navigation (NOE) (TSD) (NSD) (C)	
		Transition to Hover	Prepare and Send Digital SPOT Report (Ground Search)	
0:37:10	Hover	Hover Masked (C)	Monitor Threat (TSD)	10.4
			Select Overwatch Position (TSD)	
0:37:19				
0:38:50	Comm Receive		Receive Digital Message	
			Prepare and Send Digital SPOT Report (Ground Search)	
			Receive Digital Message	

Mission Scenario Timeline

time	Scenario summary	Pilot tasks	Copilot tasks	Segment
0:40:15	Comm Transmit		Prepare and Send Digital Free Text Message (?)	
			Select Navigation Waypoint (TSD)	
			Prepare and Send Digital Movement Report	
0:40:25	Depart OP2	Transition to NOE		
	Manuever NOE/BO	Fly NOE (C)	Monitor Threat (TSD)	10.4
		Perform Navigation (NOE) (HMD) (C)	Perform Navigation (NOE (TSD) (C)	
			Select Overwatch Position (TSD)	
		Transition to Hover		
		Establish Overwatch Position		
		Hover Masked (C)		
0:45:20	Arrive OP3	Transition to NOE	Select Observation Point (TSD)	
	Assume observation position	Fly NOE (C)	Perform Navigation (TSD) (NSD)	
		Perform Navigation (NOE) HHMD)		
		Transition to Hover		
		Hover Masked (C)		
0:45:30	Search for threat		Set Up Automatic Ground Search Configuration	12.1
		Unmask Aircraft (Search)		
		Hover Unmasked (C)		
0:45:40	Remask	Mask Aircraft		
	Threat identified as tanks etc.	Hover Masked (C)	Review Automatic Search Target Frames	
			Prepare and Send Digital Free Text Message	
0:50:25	Comm transmit		Prepare and Send Digital SPOT Report (Ground Search)	
0:51:05	Receive comm		Receive Digital Messsage	
			Select Observation Point (TSD)	10.2
			Prepare and Send Digital Free Text Message	
0:51:55	Depart OP3	Transition to NOE	Prepare and Send Digital Movement Report	
	Maneuver NOE/BO	Fly NOE (C)	Monitor Threat (TSD)	11.3
		Perform Navigation (NOE) (HMD) (C)	Perform Navigation (TSD) (NSD)	
			Set Up Automatic Ground Search Configuration	
		Transition to Hover	Review Automatic Search Target Frames	
		Establish Overwatch Position	Select Overwatch Position (TSD)	
		Hover Masked (C)	Set Up Automatic Ground Search Configuration	
		Unmask Aircraft		
		Hover Unmasked (C)	Review Automatic Search Target Frames	
0:54:35	Threat detected visually	Mask Aircraft		
	Aircraft fired at	Select Gun (??)		
		Hover Masked (C)		13.7
0:54:38	Comm transmit	Monitor HMD Target Symbology (Pilot)	Select Transmit Radio (copilot)	
		Unmask Aircraft (Attack)		
	Fires at target	Fire Gun, Ground Target, Fixed Forward	Monitor Gun Engagement	
		Evade Threat		
		Remask Aircraft		

Mission Scenario Timeline

time	Scenario summary	Pilot tasks	Copilot tasks	Segment
0:54:48	Target destroyed	Hover Masked	Select Transmit Radio (copilot)	
			Receive External Voice Communication (Copilot)	
0:55:10		Transition to NOE	Prepare and Send Digital Free Text Message	
0:55:25	Report destruction of target	Fly NOE (C)	Prepare and Send Digital Message, BDA Report (??)	
			Prepare and Send Digital Movement Report	
0:56:20	Maneuver to new observation pt		Select Observation Point (TSD)	10.2
		Perform Navigation (NOE) (HMD)	Perform Navigation (TSD) (NSD)	
		Transition to Hover		
		Hover Masked (C)	Set Up Automatic Ground Search Configuration	12.4
		Receive External Communication (Pilot)		
0:56:30	Search for threat activity	Unmask Aircraft (Search)	Perform Search, Slew Mode (SMD) (C)	
0:56:39	Remasks	Mask Aircraft	Review Automatic Search Target Frames	
		Hover Masked	Prepare and Send Digital SPOT Report (Ground Search)	
1:00:05	Comm receive		Receive Digital Movement Message	
1:00:15	Comm transmit		Prepare and Send Digital Movement Report	
	Depart OP 3a		Select Navigation Waypoint	10.1
1:00:27	Maneuver NOE/BO to OP4	Transition to NOE	Prepare and Send Digital Movement Report	
		Fly NOE	Monitor Threat	10.4
		Perform Navigation (NOE) (HMD)	Perform Navigation NOE) (TSD) (C)	
			Select Overwatch Position	
		Transition to Hover		
		Establish Overwatch Position		
		Hover Masked (C)	Monitor Threat	
		Transition to NOE	Set Up Automatic Ground Search Configuration	11.3
		Fly NOE	Review Automatic Search Target Frames	
			Select Observation Point (TSD)	
1:07:30	Arrive at OP4	Transition to Hover		
1:07:50	Comm transmit	Mask Aircraft	Prepare and Send Digital Free Text Message	

RAI JN N ANALYSIS

FUNCTION: Monitor Threat (SMD) Baseline TOTAL TIME (Approximate) Seconds

TASKS			SUBSYSTEM	WORKLOAD COMPONENTS			SWITCH DESCRIPTION	DURATION (SEC) DISCRETE/ CONTIN.	
VERB	OBJECT			SENSORY	COGNITIVE	PSYCHOMOTOR			
Press	PLTS ASE	C	Survivability/SMD (S)/(SMD)	Observe ASE Display on SMD and ASE Menu on LMPD V-1.0(I)	Verify Correct Status (ASE Mode Selected) C-3.7	Press Switch P-2.2(L)	Hard Pushbutton, SMD Bezel 115 (HSB) (On LMFD)	1.0	
Check for	Line of Bearing (LOB) Arrows	C	Survivability/SMD (S)/(SMD)	Visually Scan Display for LOB Arrows V-4.0(I)	Verify Correct Status (Area Secure) C-3.7			1.0	
Monitor	Threat Warning	C	Survivability/CWA (S)/(CWA)	Orient to Sound A-2	S- R Association C-1.0				C
Check for	Emitting Threats	C	Survivability/TSD (S)/(TSD)	Visually Scan Display for Strobing Straight Lines. Attend to Auditory Pings V-4.0(I)/A-4.3	Verify Correct Status (Area Secure) C-3.7			1.0	
Check for	Tracking, Emitting Threats	C	Survivability/TSD (S)/(TSD)	Visually Scan Display for Colored Strobing Straight Line V-4.0(I)	Verify Correct Status (Area Secure) C-3.7			1.0	

RAI IN N ANALYSIS

FUNCTION: Prepare and Send Digital Message, SPOT Report (At End of Ground Search) 5 Dec 91 (B2) TOTAL TIME (Approximate) 15.0 Seconds

TASKS			SUBSYSTEM	WORKLOAD COMPONENTS			SWITCH DESCRIPTION	DURATION (SEC) DISCRETE/ CONTIN.	
VERB	OBJECT			SENSORY	COGNITIVE	PSYCHOMOTOR			
Touch	SPTRPT	C	Target Acquisition/ Communication/ LMPD/SMD (TAS)/(C)/(LMD)/ (SMD)	Observe Inverse Video/ Highlighting. View SPOT Report Format on SMD V-1.0(I)	Verify Correct Status (SPOT Report Format Selected) C-3.7	Touch Touchscreen Menu Item P-2.2(L)	Touchfield 301.3.2 (T) (On LMPD)	.5	
Cycle	ADRS	C	Communications/ SMD (C)/(SMD)	View Options Highlighted V-1.0	Decide Correct Address Selected C-4.6	Cycle Switch (Successive Presses and Releases of Switch) P-2.2(L)	Soft Pushbutton, SMD Bezel 117.120.122.122 (SSB) (On LMFD)	2.5	
Slew	Cursor	C	Copilot Mission Grip/TSD (CMG)/(TSD)	Feel Switch Location and Movement. View Cursor Position Move to Icon/POI V-5.0 (I)	Verify Cursor is Over Icon/POI C-6.8	Manipulate Cursor Hand-Eye Coordination P-4.6 (R)	5-Position, 2-Axis Variable Transducer with Pushbutton in Center (VTB) CPG Mission Grip)	5.0	
Press and Release	Hook	C	Copilot Mission Grip/TSD (CMG)/(TSD)	View Cursor at Desired Location/Position. Observe Target Information on SPTRPT V-5.9 (I)	Judgment (Cursor at Desired Location and Desired Target Information on SPTRPT) C-6.8	Press and Release Hook Switch P-2.2 (R)	5-Position, 2-Axis Variable Transducer with Pushbutton in Center (VTB) (On CPG Mission Grip)	1.0	

RAI IN N ANALYSIS

FUNCTION: Prepare and Send Digital Message, SPOT Report (At End of Ground Search (Continued 1) TOTAL TIME (Approximate) Seconds

TASKS			SUBSYSTEM	WORKLOAD COMPONENTS			SWITCH DESCRIPTION	DURATION (SEC) DISCRETE/ CONTIN.	
VERB	OBJECT			SENSORY	COGNITIVE	PSYCHOMOTOR			
Cycle	MY ACTVTY	C	Communication/ SMD (C)/(SMD)	View Options Highlighted. Read Activities V-5.9(I)	Decide Correct Activity Selected C-3.7	Cycle Switch (Successive Presses and Release of Switch) P-2.2(L)	Soft Pushbutton, SMD Bezel 117.120.122.121 (SSB) (On LMFD)	2.5	
Touch	SND ROUT	C	Communication/ SMD (C)/(SMD)	Observe Inverse Video/ Highlighting. Read Message Status on SMD V-5.9(I)	Verify Correct Status (Send Routine Selected) C-3.7	Touch Touchscreen Menu Item P-2.2(L)	Touchfield 117.120.122.8 (T) (On LMPD)	.5	

Scenario Functions and Display Item Sequences

Function	Display Item Sequence
Monitor Threat (TSD)	map threat symbology display bearing to threat bearing to emitting threat brg to tracking, emitting threat
Perform Before Takeoff Check (Copilot)	fuel quantity WCA counts checklist:Before TakeOff button:SAVE&EXIT
Perform Navigation (Contour) (TSD) (C)	flight plan map button:NAVOLY WPs and legs flight path:actual button:FARPOLY FARP overlay
Perform Navigation (NOE (TSD) (C)	flight plan map button:NAVOLY position:threats position:friendlies flight path:actual button:7.5K map:7.5K scale
Prep & Send Digital Free Text Message	button:FreeText menu:addressees characters typed into Free Text button:SEND ROUT
Prep & Send Digital Message, BDA Report	button:BDA menu:addressees menu:coverage menu:targets destroyed characters typed into BDA "Starttime"+typed characters "Endtime"+typed characters button:SEND ROUT
Prep & Send Digital Movement Report	button:MOVCMD menu:addressees menu:tasks menu:location menu:When

Scenario Functions and Display Item Sequences

Function	Display Item Sequence
	menu:DTG characters typed into MOVCMD button:SEND ROUT
Prep & Send Digital SPOT Report (GdSrch)	button:SPOT menu:addressees menu:my activity information:target button:SEND ROUT
Receive Digital Message	indicator:MESGS message list (INBOX) message:text map button:SAVE&EXIT
Receive Digital Movement Message	indicator:MESGS message list (INBOX) message:text button:WILCO button:SAVE&EXIT
Set Up & Review Auto Search	scan pattern graphic performance characteristics button:AZ axis button:SAVE & RETURN button:REVIEW search frame button:BROWSE button:AUTO button:DELAY 5 search frame button:NO TGT button:SAVE & RETURN
Select Navigation Waypoint (TSD)	map WPs button:SAVE&EXIT
Select Observation Point (TSD)	map TAC overlay position:threats position:friendlies position:OPs position:BP button:LOS lines of sight button:SAVE&RETURN

Scenario Functions and Display Item Sequences

Function	Display Item Sequence
Select Overwatch Position (TSD)	map
Select Transmit Radio (copilot)	list:radios
	button:XMIT 1
	button:XMIT 2
	button:XMIT 3
	button:XMIT 4
	button:XMIT 5
	button:XMIT PWR
	button:SQL 1
	button:SQL 2
	button:SQL 3
	button:SQL 4
	button:SQL 5
	button:SAVE&EXIT
Respond to Advisory Alert (Stored)	WCA counts
	List:WCA's
	Info on Alerts
	Checklist:ADVS PROC
	button:CHECK
	button:SAVE&EXIT
Respond to Caution Alert (Stored)	WCA counts
	List:WCA's
	Info on Alerts
	Checklist:EMERG
	button:CHECK
	button:SAVE&EXIT
Resp to Warn: Auto Flt Cont Sys, Nonrecov Fail	WARNING banner
	List:WCA's
	Info on Alerts
	Checklist:EMERG
	button:CHECK
	button:SAVE&EXIT
	list:radios
	button:XMIT 1
	button:XMIT 2
	button:XMIT 3
	button:XMIT 4
	button:XMIT 5
	button:SAVE&EXIT
Resp to Warn: Auto Flt Cont Sys, Recov Fail	WARNING banner
	List:WCA's
	Info on Alerts

Scenario Functions and Display Item Sequences

Function	Display Item Sequence
Respond to Warning: Engine Fire (L or R) (copilot)	WARNING banner
	List:WCA's
	Info on Alerts
	EngR ON/OFF status
	EngR Oil temp
	EngR Oil pressure
	EngR Turbine Gas temp
	EngR Gas Generator Turbine Speed
	EngR Torque
	EngR Power Turbine Speed
	Rotor Speed
	EngL ON/OFF status
	EngL Oil temp
	EngL Oil pressure
	EngL Turbine Gas temp
	EngL Gas Generator Turbine Speed
	EngL Torque
	EngL Power Turbine Speed
	Total Fuel Flow
	R Fuel Flow
	L Fuel Flow
	Main Fuel Quantity
	R Aux Fuel Quantity
	L Aux Fuel Quantity
	MGB Oil Temp
	MGB Oil Pressure
	Checklist:EMERG
	button:CHECK
	button:SAVE&EXIT
	list:radios
	button:XMIT 1
	button:XMIT 2
	button:XMIT 3
	button:XMIT 4
	button:XMIT 5
	button:SAVE&EXIT
	EngR ON/OFF status
	EngR Oil temp
	EngR Oil pressure
	EngR Turbine Gas temp
	EngR Gas Generator Turbine Speed
	EngR Torque
	EngR Power Turbine Speed
	Rotor Speed
	EngL ON/OFF status
	EngL Oil temp
	EngL Oil pressure
	EngL Turbine Gas temp
	EngL Gas Generator Turbine Speed
	EngL Torque
	EngL Power Turbine Speed

Scenario Functions and Display Item Sequences

Function	Display Item Sequence
	Total Fuel Flow R Fuel Flow L Fuel Flow Main Fuel Quantity R Aux Fuel Quantity L Aux Fuel Quantity MGB Oil Temp MGB Oil Pressure
Resp to Warn: EngineOut(LorR),Inflt,Nonrecov(copilot)	WARNING banner List:WCA's Info on Alerts Checklist:EMERG button:CHECK button:SAVE&EXIT list:radios button:XMIT 1 button:XMIT 2 button:XMIT 3 button:XMIT 4 button:XMIT 5 button:SAVE & EXIT
Resp to Warn: EngineOut(LorR),Inflt,Recov(copilot)	WARNING banner List:WCA's Info on Alerts EngR ON/OFF status EngR Oil temp EngR Oil pressure EngR Turbine Gas temp EngR Gas Generator Turbine Speed EngR Torque EngR Power Turbine Speed Rotor Speed EngL ON/OFF status EngL Oil temp EngL Oil pressure EngL Turbine Gas temp EngL Gas Generator Turbine Speed EngL Torque EngL Power Turbine Speed Total Fuel Flow R Fuel Flow L Fuel Flow Main Fuel Quantity R Aux Fuel Quantity L Aux Fuel Quantity MGB Oil Temp MGB Oil Pressure Checklist:EMERG button:CHECK button:SAVE&EXIT

Scenario Functions and Display Item Sequences

Function	Display Item Sequence
Resp to Warn: Prim Flt Cont Sys, Nonrecov Fail(copilot)	WARNING banner List:WCA's Info on Alerts Checklist:EMERG button:CHECK button:SAVE&EXIT list:radios button:XMIT 1 button:XMIT 2 button:XMIT 3 button:XMIT 4 button:XMIT 5 button:SAVE&EXIT
Respond to Warning: SPU Fire	WARNING banner List:WCA's Info on Alerts Checklist:EMERG button:CHECK button:SAVE&EXIT list:radios button:XMIT 1 button:XMIT 2 button:XMIT 3 button:XMIT 4 button:XMIT 5 button:SAVE&EXIT
Resp to Warning: Weapons Bay Fire, (L/R)	WARNING banner List:WCA's Info on Alerts Checklist:EMERG button:CHECK button:SAVE&EXIT list:radios button:XMIT 1 button:XMIT 2 button:XMIT 3 button:XMIT 4 button:XMIT 5 button:SAVE&EXIT
Review Advisories	WCA count List:WCA's Info on Alerts button:SAVE&EXIT
Review Cautions	WCA count List:WCA's Info on Alerts button:SAVE&EXIT

Display Item Areas

Display Item	Area
"Endtime"+typed characters	4.2
"Starttime"+typed characters	4.2
button:7.5K	3.0
button:AUTO	3.0
button:AZ axis	3.0
button:BDA	3.0
button:BROWSE	3.0
button:DELAY 5	3.0
button:FARPOLY	3.0
button:FreeText	3.0
button:LOS	3.0
button:MOVCMD	3.0
button:NAVOLY	3.0
button:NO TGT	3.0
button:REVIEW	3.0
button:SPOT	3.0
button:SQL 1	3.0
button:SQL 2	3.0
button:SQL 3	3.0
button:SQL 4	3.0
button:SQL 5	3.0
button:WILCO	3.0
button:XMIT 1	3.0
button:XMIT 2	3.0
button:XMIT 3	3.0
button:XMIT 4	3.0
button:XMIT 5	3.0
button:XMIT PWR	3.0
characters typed into BDA report	3.0
characters typed into Free Text report	33.0
characters typed into MOVCMD report	3.0
Checklist:ADVS PROC	70.0
Checklist:Before TO	70.0
Checklist:EMERG	70.0
EngL Gas Generator Turbine Speed	3.5
EngL Oil pressure	3.5
EngL Oil temp	3.5
EngL ON/OFF status	3.5
EngL Power Turbine Speed	3.5
EngL Torque	3.5
EngL Turbine Gas temp	3.5
EngR Gas Generator Turbine Speed	3.5
EngR Oil pressure	3.5
EngR Oil temp	3.5
EngR ON/OFF status	3.5
EngR Power Turbine Speed	3.5
EngR Torque	3.5

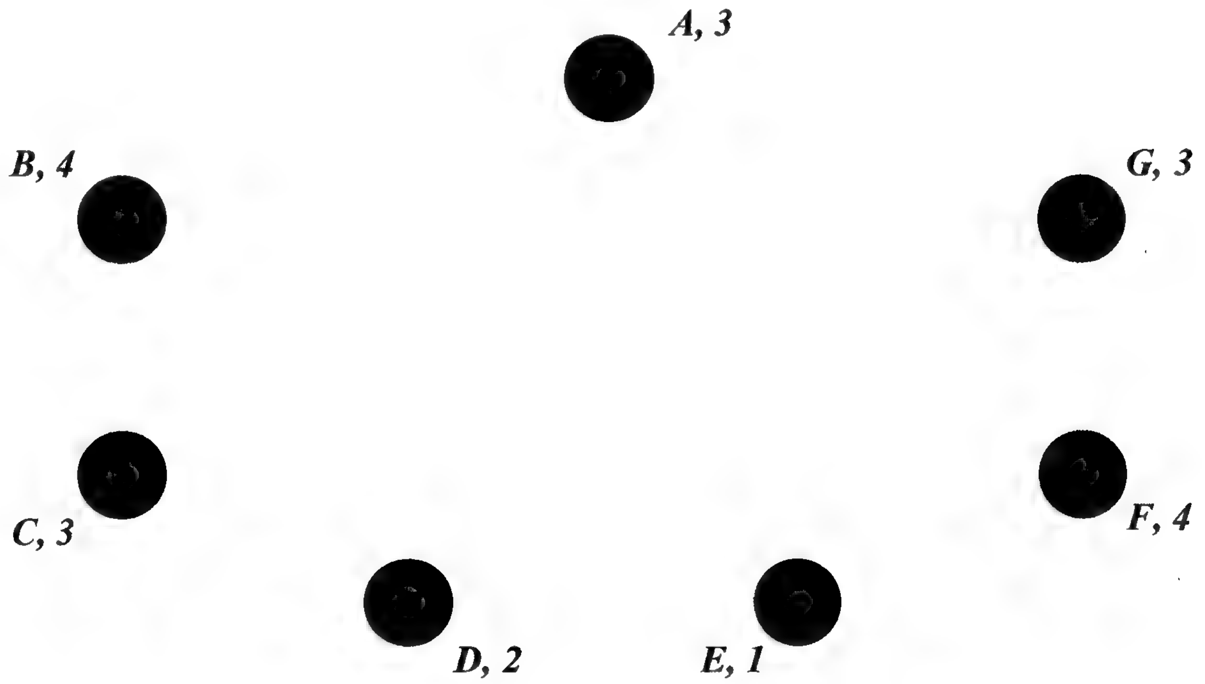
Display Item Areas

Display Item	Area
EngR Turbine Gas temp	3.5
flight plan	70.0
Fuel Quantity	3.5
indicator:MESGS	3.0
Info on Alerts	63.0
information:target	33.0
L Aux Fuel Quantity	3.5
L Fuel Flow	3.5
list:radios	68.0
List:WCA's	63.0
Main Fuel Quantity	3.5
map	82.0
map:7.5K scale	82.0
menu:coverage	10.0
menu:DTG	10.0
menu:location	10.0
menu:my activity	10.0
menu:targets destroyed	10.0
menu:tasks	10.0
menu:when	10.0
message list (INBOX)	60.0
message text	60.0
MGB Oil Pressure	3.5
MGB Oil Temp	3.5
performance characteristics	42.0
R Aux Fuel Quantity	3.5
R Fuel Flow	3.5
Rotor Speed	3.5
scan pattern graphic	40.0
search frame	82.0
threat symbology display	60.0
Total Fuel Flow	3.5
WARNING banner	16.0
WCA count	2.0
button:Send Free Text Message	3.0
button:Send BDA Report	3.0
button:Send MOVCMD Report	3.0
button:Send SPOT Report	3.0
menu:Addressees for Free Text Msg	10.0
menu:Addressees for BDA Report	10.0
menu:Addressees for MOVCMD Report	10.0
menu:Addressees for SPOT Report	10.0
button:Save Message	3.0
button:Save WPs	3.0
button:Save OP	3.0
button:Check ADV PROC	3.0
button:Check EMERG PROC	3.0

Appendix D

Transformation Procedure Examples

Placing Initial Nodes

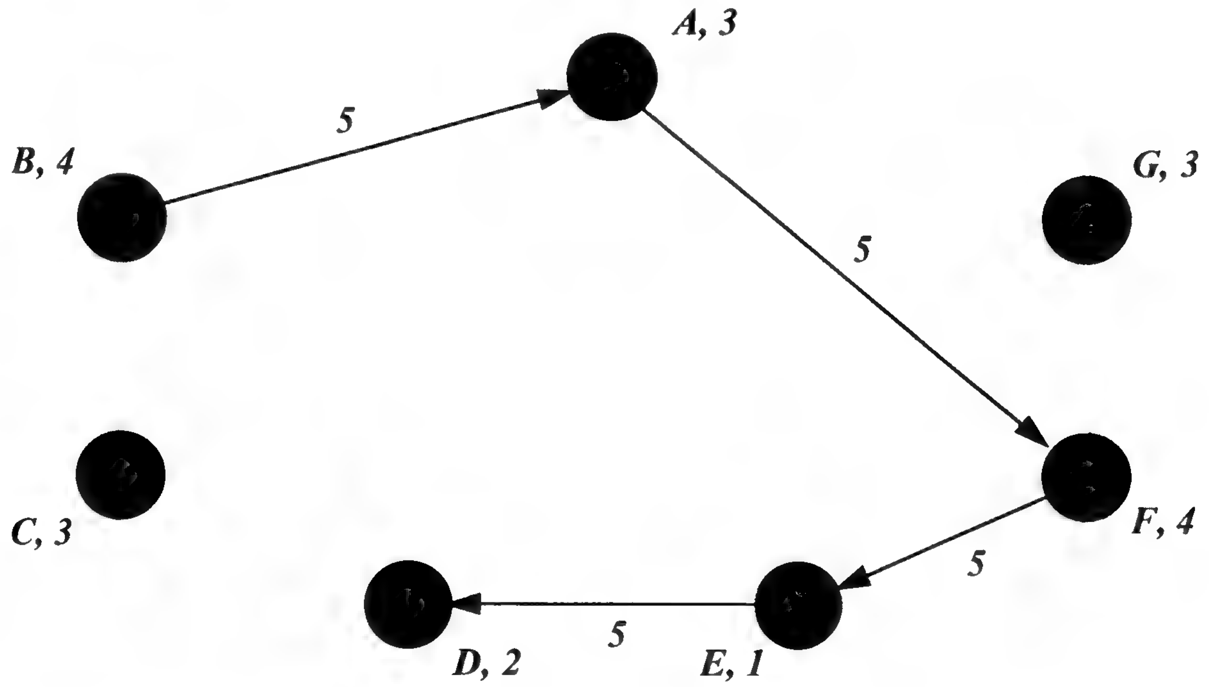


PAGE SIZE = 7

LINK SIZE = 1

- Display items, with identifiers and areas

Placing the First Sequence

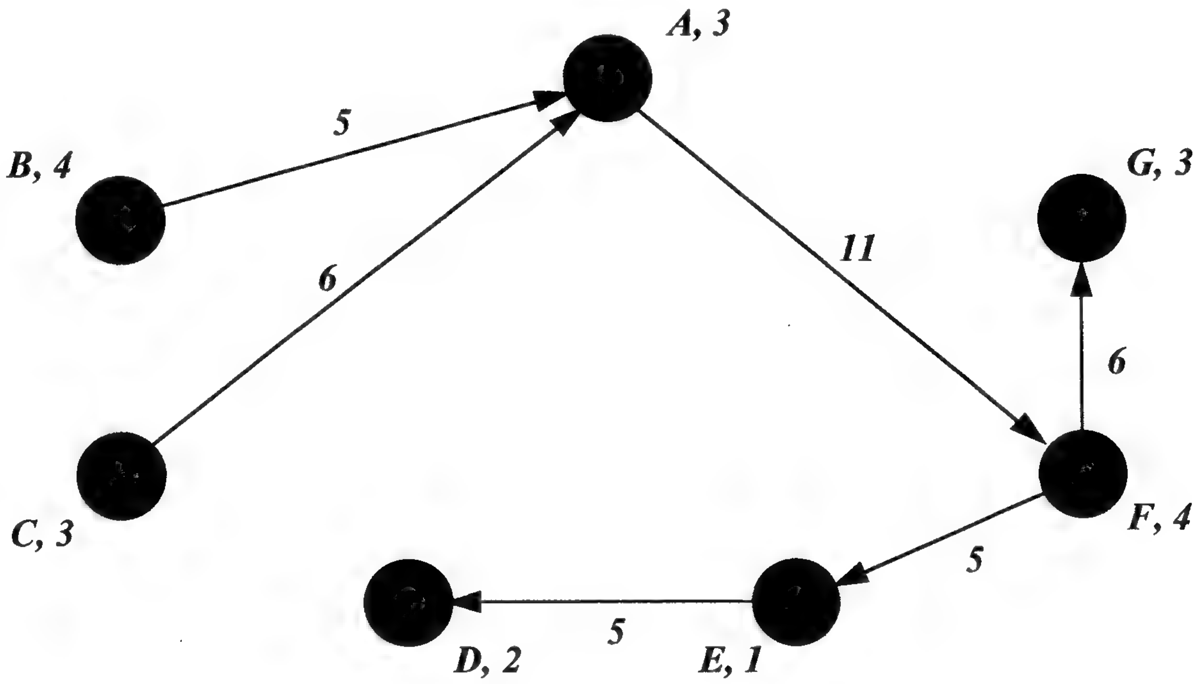


PAGE SIZE = 7

LINK SIZE = 1

- Incorporating access sequence [(B, A, F, E, D), 5, 1]

Placing the Second Sequence

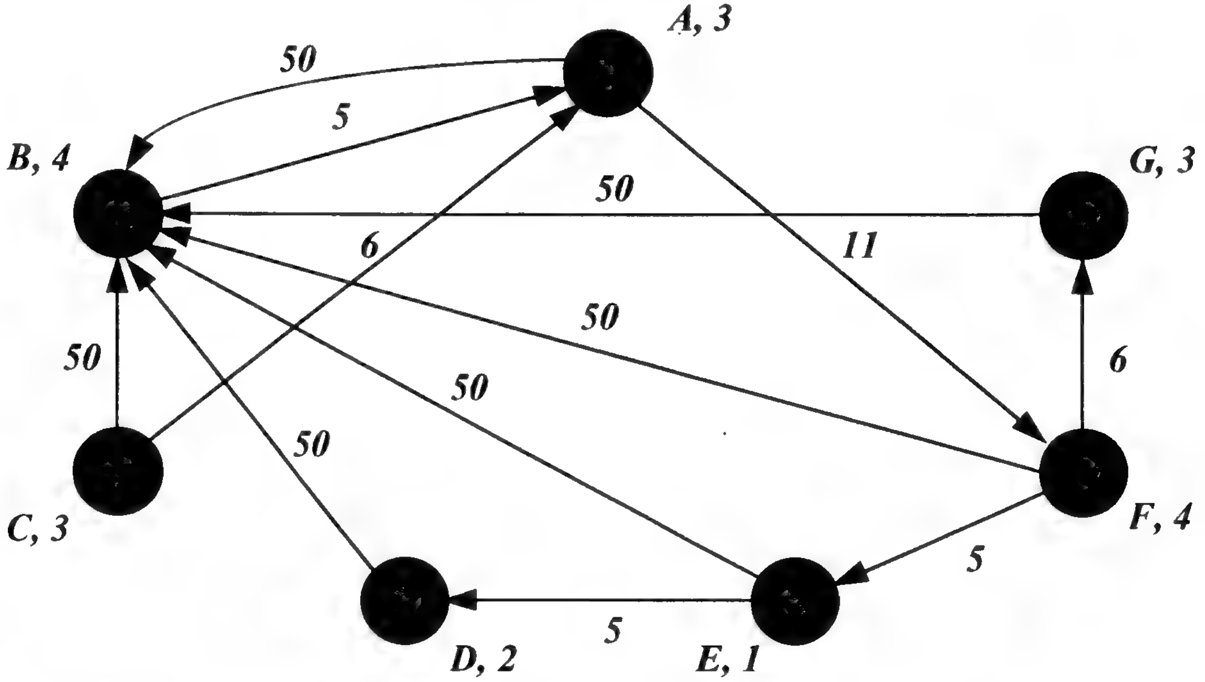


PAGE SIZE = 7

LINK SIZE = 1

- Incorporating access sequence [(C, A, F, G), 3, 2]

Adding an Always-Accessible Node

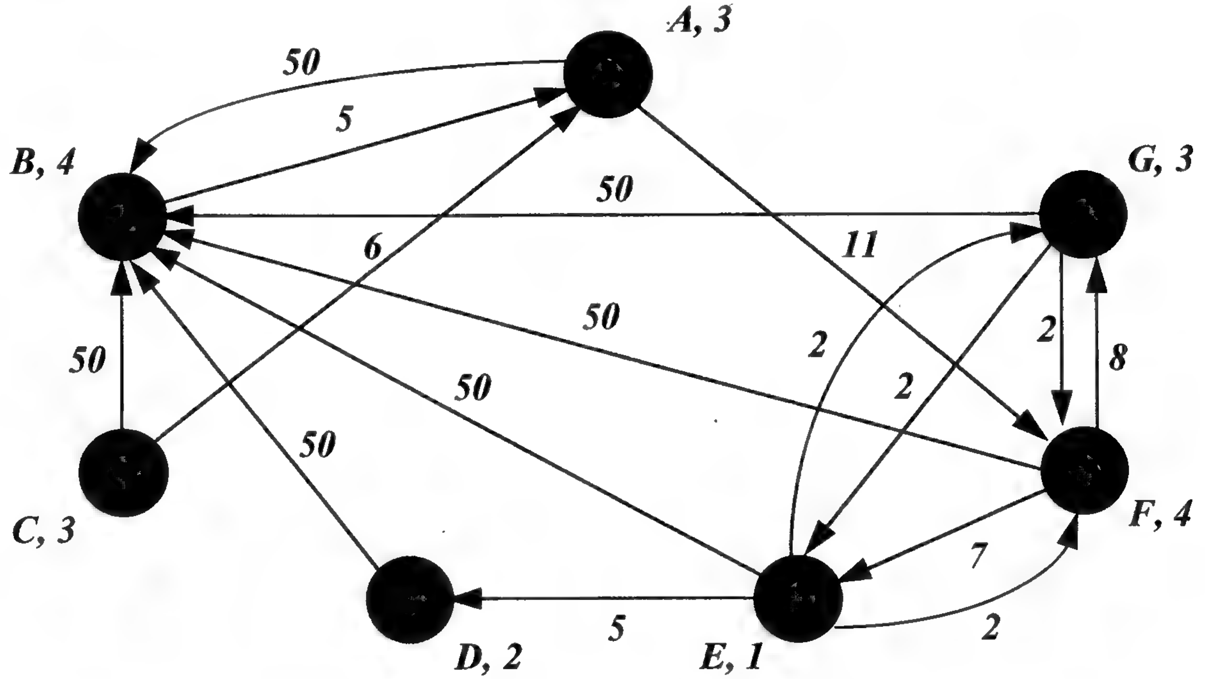


PAGE SIZE = 7

LINK SIZE = 1

- Display-item B should always be accessible

Adding a Conceptual Cluster

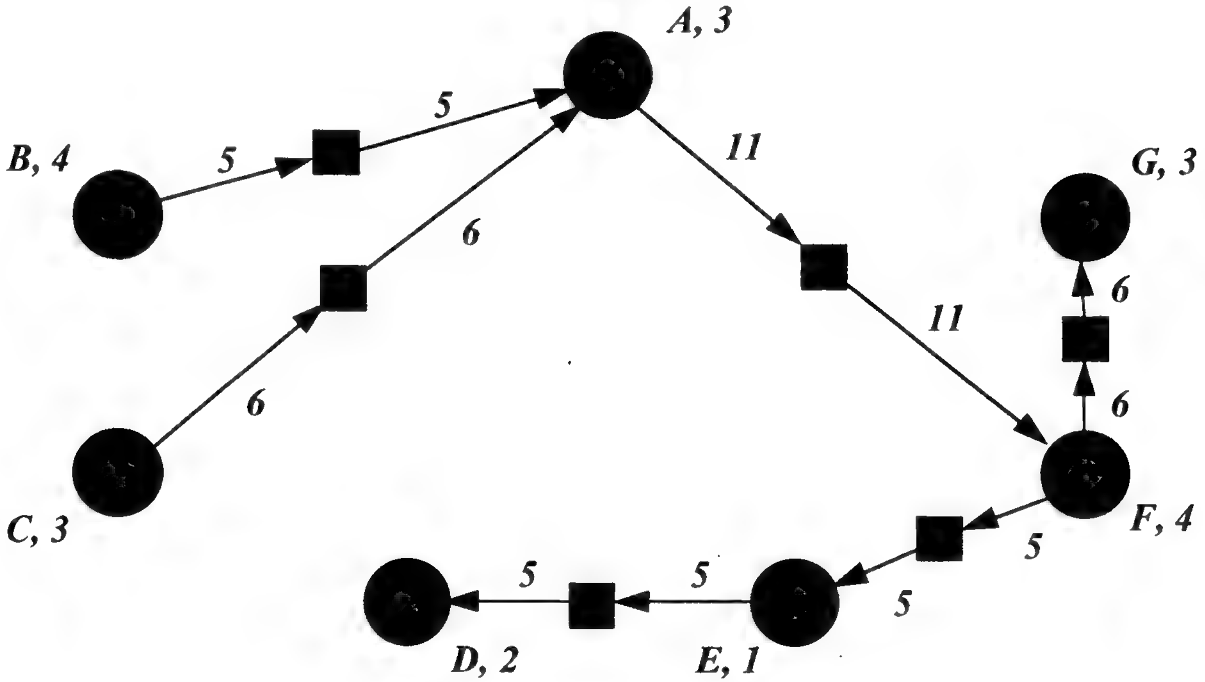


PAGE SIZE = 7

LINK SIZE = 1

- [{E, F, G}, 2] forms a conceptual cluster

Adding Stepping-Stone Nodes

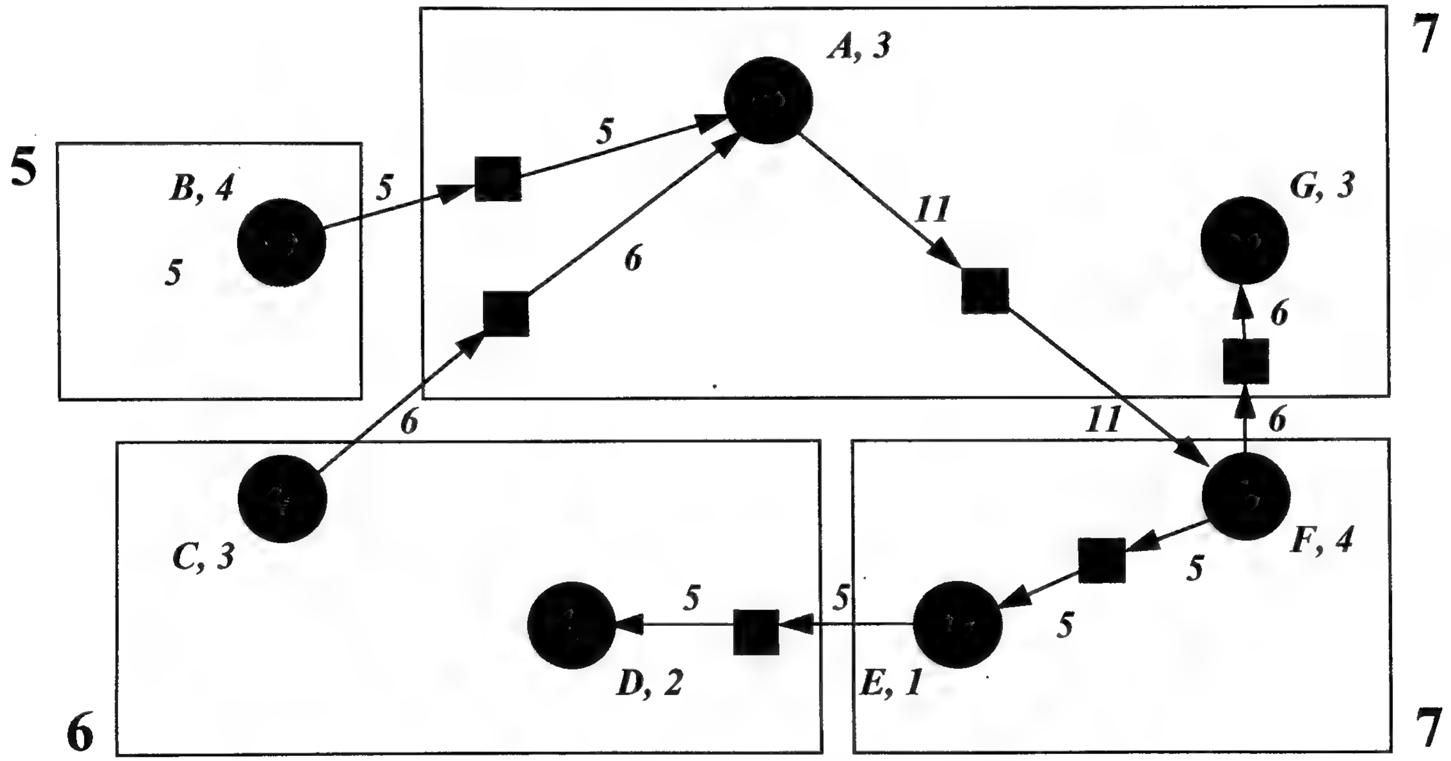


PAGE SIZE = 7

LINK SIZE = 1

- *Stepping-stone nodes* are introduced on each edge

A Good Partition

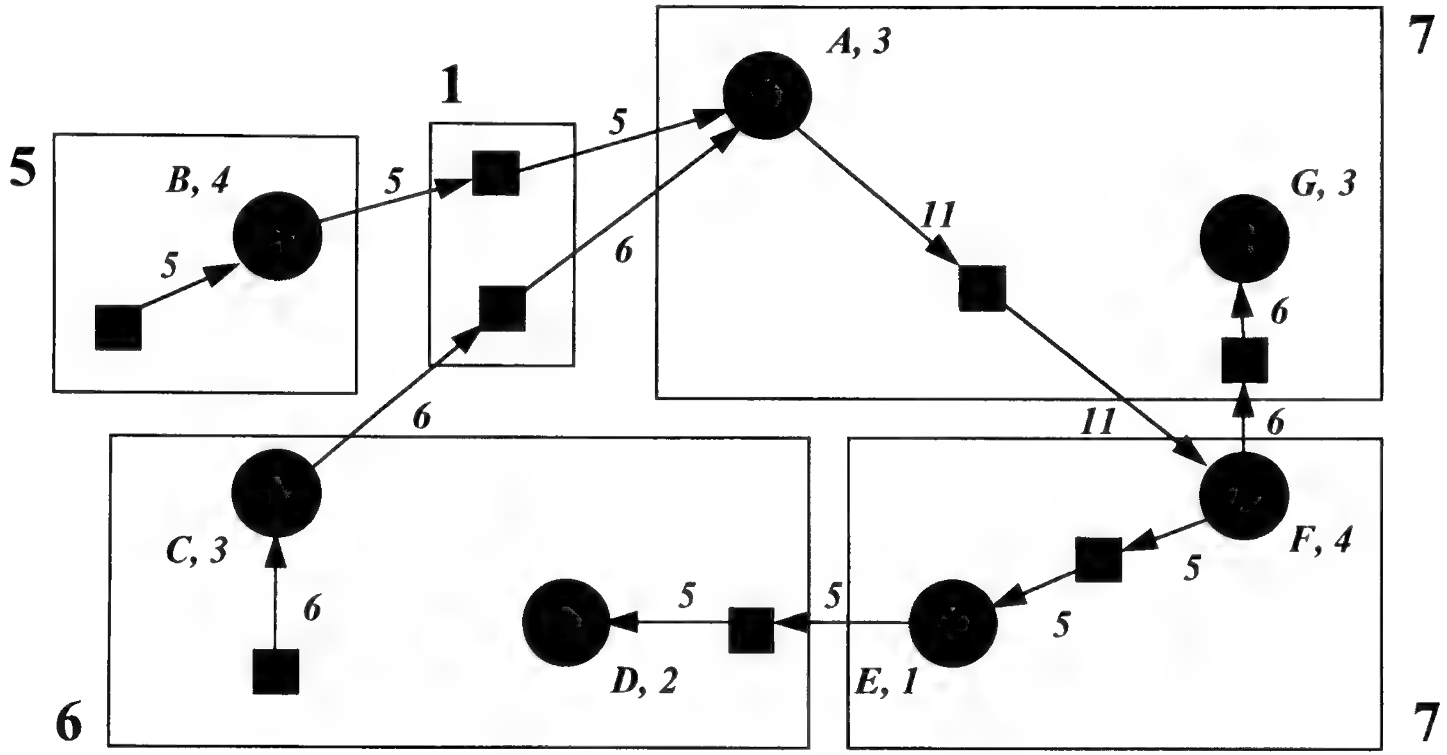


PAGE SIZE = 7

LINK SIZE = 1

- Partitioning the graph into pages: cut-set cost = 33

A Bad Partition

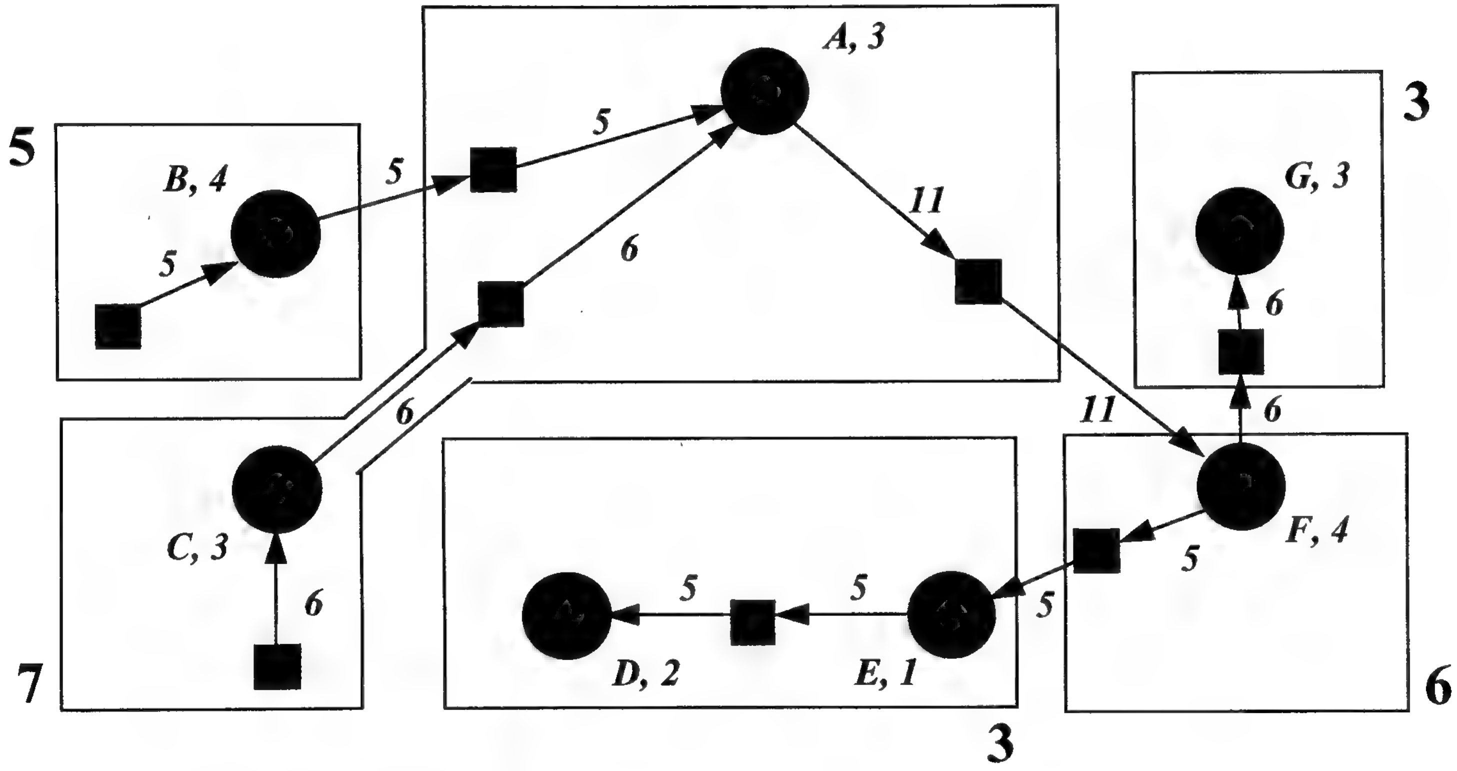


PAGE SIZE = 7

LINK SIZE = 1

- An alternative partitioning: cut-set cost = 44

The Computer-Generated Partition



PAGE SIZE = 7

LINK SIZE = 1

- Cut-set cost = 27

Appendix E

Documentation of Results

Page Organization Results for Comanche Data Script.3

```
Page 0
        Total area occupied for page 0:         63.00
                    140     Checklist: Before TO
                    Connects to page 11

Page 1
        Total area occupied for page 1:         84.00
                    106     button: 7.5K
                    113     button: FARPOLY
                    115     button: LOS
                    117     button: NAVOLY
                    172     Map
                    Connects to page 8
                    Connects to page 11
                    Connects to page 12
                    Connects to page 18

Page 2
        Total area occupied for page 2:         66.00
                    139     Checklist: ADVS PROC
                    218     button: Check ADVS PROC
                    Connects to page 11

Page 3
        Total area occupied for page 3:         59.00
                    138     char typed MOVCMD
                    177     menu: DTG
                    178     menu: location
                    181     menu: tasks
                    182     menu: when
                    209     button: Send MOVCMD Report
                    213     menu: addressees MOVCMD
                    Connects to page 11

Page 4
        Total area occupied for page 4:         66.00
                    169     list: Radios
                    Connects to page 11
                    Connects to page 17

Page 5
        Total area occupied for page 5:         66.00
                    183     message list (INBOX)
                    Connects to page 11
                    Connects to page 15

Page 7
        Total area occupied for page 7:         27.00
                    119     button: REVIEW
                    121     button: SAVE & RETURN
```

124	button: SQL1
125	button: SQL2
126	button: SQL3
127	button: SQL4
128	button: SQL5

Connects to page 11
Connects to page 19

Page 8

Total area occupied for page 8: 55.00

137	char typed Free Text
163	indicator: MESGS
207	button: Send Free Text
211	menu: addressees Free Text

Connects to page 5
Connects to page 11

Page 9

Total area occupied for page 9: 68.40

101	"Endtime"
102	"Starttime"
109	button: BDA
114	button: Free Text
116	button: MOVCMD
123	button: SPOT
136	char typed BDA
176	menu: coverage
180	menu: targets destroyed
208	button: Send BDA Report
212	menu: addressees BDA

Connects to page 3
Connects to page 8
Connects to page 11
Connects to page 17

Page 10

Total area occupied for page 10: 72.00

164	Info on Alerts

Connects to page 2
Connects to page 11
Connects to page 22
Connects to page 23

Page 11

Total area occupied for page 11: 99.50

129	button: WILCO
161	Fuel Quantity
170	List: WCAs
205	Warning Banner
206	WCA counts
215	button: Save Message
216	button: Save WPs
217	button: Save OPs

Connects to page 0

Connects to page 10

Page 12

Total area occupied for page 12: 63.00

173 Map 7.5K Scale
Connects to page 11

Page 13

Total area occupied for page 13: 91.00

108 button: AZ Axis
189 Performance Characteristics
198 Scan Pattern Graphic
Connects to page 7
Connects to page 11

Page 15

Total area occupied for page 15: 66.00

184 Message Text
Connects to page 1
Connects to page 11

Page 16

Total area occupied for page 16: 66.00

159 Flight Plan
Connects to page 1
Connects to page 11

Page 17

Total area occupied for page 17: 83.00

130 button: XMIT1
131 button: XMIT2
132 button: XMIT3
133 button: XMIT4
134 button: XMIT5
135 button: XMIT PWR
165 Info on Target
179 menu: my activity
210 button: Send SPOT Report
214 menu: addressees SPOT
Connects to page 7
Connects to page 11
Connects to page 22

Page 18

Total area occupied for page 18: 63.00

201 Threat Symbology
Connects to page 11

Page 19

Total area occupied for page 19: 75.00

107 button: AUTO
110 button: BROWSE
112 button: DELAY 5
118 button: NO TGT

199	Search Frame

Connects to page 11

Page 22

Total area occupied for page 22: 86.50

143	EngL Gas Gen Turbine Speed
144	EngL Oil Pressure
145	EngL Oil Temp
146	EngL ON/OFF
147	EngL Power Turbine Speed
148	EngL Torque
149	EngL Turbine Gas Temp
150	EngR Gas Gen Turbine Speed
151	EngR Oil Pressure
152	EngR Oil Temp
153	EngR ON/OFF
154	EngR Power Turbine Speed
155	EngR Torque
156	EngR Turbine Gas Temp
166	L Aux Fuel Quantity
167	L Fuel Flow
171	Main Fuel Flow
185	MGB Oil Press
186	MGB Oil Temp
195	R Aux Fuel Quantity
196	R Fuel Flow
197	Rotor Speed
203	Total Fuel Flow

Connects to page 11

Connects to page 23

Page 23

Total area occupied for page 23: 69.00

141	Checklist: EMERG
219	button: Check EMERG

Connects to page 4

Connects to page 11

Results from Optimization Run on Comanche Data Script.3

Input Data					Results/Analysi	
Function	Display Item Sequence	Area	Freq	Criticality	Page Assigned	Button presses/function
Monitor Threat			11	8.5		
	map	60			1	
	threat symbology display	60			18	1
Perform Before Takeoff Check			1	10		
	fuel quantity	3			11	
	WCA counts	3			11	0
	checklist:Before TO	60			0	1
Perform Navigation (Contour)			2	6		
	flight plan	60			16	
	map	60			1	1
	button:NAVOLY	3			1	0
	button:FARPOLY	3			1	0
Perform Navigation (NOE)			12	7		
	flight plan	60			16	
	map	60			1	1
	button:NAVOLY	3			1	0
	button:7.5K	3			1	0
	map:7.5K scale	60			12	1
Prep & Send Digital Free Text Message			6	2		
	button:FreeText	3			9	
	menu:addressees for Free Text	10			8	1
	characters typed into Free Text	33			8	0
	button:SEND Free Text Msg	3			8	0
Prep & Send Digital Msg, BDA Rept			1	3		
	button:BDA	3			9	
	menu:addressees for BDA Repo	10			9	0
	menu:coverage	10			9	0
	menu:targets destroyed	10			9	0
	characters typed into BDA	3			9	0
	"Starttime"+typed characters	4.2			9	0
	"Endtime"+typed characters	4.2			9	0
	button:SEND BDA Report	3			9	0
Prep & Send Digital Movement Rept			7	1.5		
	button:MOVCMD	3			9	
	menu:addressees for MOVCMD I	10			3	1
	menu:tasks	10			3	0
	menu:location	10			3	0
	menu:When	10			3	0
	menu:DTG	10			3	0
	characters typed into MOVCMD	3			3	0
	button:SEND MOVCMD Report	3			3	0
Prep & Send Digital SPOT Rept (GdSrch)			4	4		
	button:SPOT	3			9	
	menu:addressees for SPOT Rep	10			17	1
	menu:my activity	10			17	0
	information:target	33			17	0
	button:SEND SPOT Report	3			17	0

Results from Optimization Run on Comanche Data Script.3

Input Data					Results/Analysi	
Receive Digital Message			4	0.5		
	indicator:MESGS	3			8	
	message list (INBOX)	60			5	1
	message:text	60			15	1
	map	60			1	1
	button:SAVE Message	3			11	1
Receive Digital Movement Message			2	1.5		
	indicator:MESGS	3			8	
	message list (INBOX)	60			5	1
	message:text	60			15	1
	button:WILCO	3			11	1
	button:SAVE Message	3			11	0
Set Up & Review Auto Search			8	5		
	scan pattern graphic	40			13	
	performance characteristics	42			13	0
	button:AZ axis	3			13	0
	button:SAVE & RETURN	3			7	1
	button:REVIEW	3			7	0
	search frame	60			19	1
	button:BROWSE	3			19	0
	button:AUTO	3			19	0
	button:DELAY 5	3			19	0
	search frame	60			19	0
	button:NO TGT	3			19	0
Select Navigation Waypoint			4	6		
	map	60			1	
	button:SAVE WPs	3			11	1
Select Observation Point			5	6		
	map	60			1	
	button:LOS	3			1	0
	button:SAVE OP	3			11	1
Select Overwatch Position			6	7		
	map	60			1	
Select Transmit Radio			2	4		
	list:radios	60			4	
	button:XMIT 1	3			17	1
	button:XMIT 2	3			17	0
	button:XMIT 3	3			17	0
	button:XMIT 4	3			17	0
	button:XMIT 5	3			17	0
	button:XMIT PWR	3			17	0
	button:SQL 1	3			7	1
	button:SQL 2	3			7	0
	button:SQL 3	3			7	0
	button:SQL 4	3			7	0
	button:SQL 5	3			7	0
Respond to Advisory Alert (Stored)			1	90		
	WCA counts	3			11	
	List:WCA's	60			11	0

Results from Optimization Run on Comanche Data Script.3

Input Data					Results/Analysi	
	Info on Alerts	60			10	1
	Checklist:ADVS PROC	60			2	1
	button:CHECK ADVS PROC	3			2	0
Respond to Caution Alert (Stored)			1	95		
	WCA counts	3			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0
Resp to Warn: Auto Flt Cont Sys, Nonrecov Fail			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0
	list:radios	60			4	1
	button:XMIT 1	3			17	1
	button:XMIT 2	3			17	0
	button:XMIT 3	3			17	0
	button:XMIT 4	3			17	0
	button:XMIT 5	3			17	0
Resp to Warn: Auto Flt Cont Sys, Recov Fail			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
Respond to Warning: Engine Fire			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	EngR ON/OFF status	3.5			22	1
	EngR Oil temp	3.5			22	0
	EngR Oil pressure	3.5			22	0
	EngR Turbine Gas temp	3.5			22	0
	EngR Gas Generator Turbine Spe	3.5			22	0
	EngR Torque	3.5			22	0
	EngR Power Turbine Speed	3.5			22	0
	Rotor Speed	3.5			22	0
	EngL ON/OFF status	3.5			22	0
	EngL Oil temp	3.5			22	0
	EngL Oil pressure	3.5			22	0
	EngL Turbine Gas temp	3.5			22	0
	EngL Gas Generator Turbine Spe	3.5			22	0
	EngL Torque	3.5			22	0
	EngL Power Turbine Speed	3.5			22	0
	Total Fuel Flow	3.5			22	0
	R Fuel Flow	3.5			22	0
	L Fuel Flow	3.5			22	0
	Main Fuel Quantity	3.5			22	0
	R Aux Fuel Quantity	3.5			22	0
	L Aux Fuel Quantity	3.5			22	0
	MGB Oil Temp	3.5			22	0
	MGB Oil Pressure	3.5			22	0
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0

Results from Optimization Run on Comanche Data Script.3

Input Data					Results/Analysi	
	list:radios	60			4	1
	button:XMIT 1	3			17	1
	button:XMIT 2	3			17	0
	button:XMIT 3	3			17	0
	button:XMIT 4	3			17	0
	button:XMIT 5	3			17	0
	EngR ON/OFF status	3.5			22	1
	EngR Oil temp	3.5			22	0
	EngR Oil pressure	3.5			22	0
	EngR Turbine Gas temp	3.5			22	0
	EngR Gas Generator Turbine Spe	3.5			22	0
	EngR Torque	3.5			22	0
	EngR Power Turbine Speed	3.5			22	0
	Rotor Speed	3.5			22	0
	EngL ON/OFF status	3.5			22	0
	EngL Oil temp	3.5			22	0
	EngL Oil pressure	3.5			22	0
	EngL Turbine Gas temp	3.5			22	0
	EngL Gas Generator Turbine Spe	3.5			22	0
	EngL Torque	3.5			22	0
	EngL Power Turbine Speed	3.5			22	0
	Total Fuel Flow	3.5			22	0
	R Fuel Flow	3.5			22	0
	L Fuel Flow	3.5			22	0
	Main Fuel Quantity	3.5			22	0
	R Aux Fuel Quantity	3.5			22	0
	L Aux Fuel Quantity	3.5			22	0
	MGB Oil Temp	3.5			22	0
	MGB Oil Pressure	3.5			22	0
Resp to Warn: EngineOut,Inflt,Nonrecov			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0
	list:radios	60			4	1
	button:XMIT 1	3			17	1
	button:XMIT 2	3			17	0
	button:XMIT 3	3			17	0
	button:XMIT 4	3			17	0
	button:XMIT 5	3			17	0
Resp to Warn: EngineOut,Inflt,Recov			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	EngR ON/OFF status	3.5			22	1
	EngR Oil temp	3.5			22	0
	EngR Oil pressure	3.5			22	0
	EngR Turbine Gas temp	3.5			22	0
	EngR Gas Generator Turbine Spe	3.5			22	0
	EngR Torque	3.5			22	0
	EngR Power Turbine Speed	3.5			22	0
	Rotor Speed	3.5			22	0
	EngL ON/OFF status	3.5			22	0
	EngL Oil temp	3.5			22	0
	EngL Oil pressure	3.5			22	0

Results from Optimization Run on Comanche Data Script.3

Input Data					Results/Analysi	
	EngL Turbine Gas temp	3.5			22	0
	EngL Gas Generator Turbine Spe	3.5			22	0
	EngL Torque	3.5			22	0
	EngL Power Turbine Speed	3.5			22	0
	Total Fuel Flow	3.5			22	0
	R Fuel Flow	3.5			22	0
	L Fuel Flow	3.5			22	0
	Main Fuel Quantity	3.5			22	0
	R Aux Fuel Quantity	3.5			22	0
	L Aux Fuel Quantity	3.5			22	0
	MGB Oil Temp	3.5			22	0
	MGB Oil Pressure	3.5			22	0
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0
Resp to Warn: Prim Flt Cont Sys, Nonrecov Fail			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0
	list:radios	60			4	1
	button:XMIT 1	3			17	1
	button:XMIT 2	3			17	0
	button:XMIT 3	3			17	0
	button:XMIT 4	3			17	0
	button:XMIT 5	3			17	0
Respond to Warning: SPU Fire			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0
	list:radios	60			4	1
	button:XMIT 1	3			17	1
	button:XMIT 2	3			17	0
	button:XMIT 3	3			17	0
	button:XMIT 4	3			17	0
	button:XMIT 5	3			17	0
Resp to Warning: Weapons Bay Fire			1	100		
	WARNING banner	16			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	Checklist:EMERG	60			23	1
	button:CHECK EMERG PROC	3			23	0
	list:radios	60			4	1
	button:XMIT 1	3			17	1
	button:XMIT 2	3			17	0
	button:XMIT 3	3			17	0
	button:XMIT 4	3			17	0
	button:XMIT 5	3			17	0
Review Advisories			1	80		
	WCA count	2			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1

Results from Optimization Run on Comanche Data Script.3

Input Data					Results/Analysi	
Review Cautions			1	90		
	WCA count	2			11	
	List:WCA's	60			11	0
	Info on Alerts	60			10	1
	Clusters		1000			
	map				15	
	button:NAVOLY				17	
	button:FARPOLY				17	
			1000			
	map				15	
	button:NAVOLY				17	
	button:7.5K				17	
			1000			
	menu:addressees for Free Text				7	
	characters typed into Free Text				17	
	button:SEND Free Text Msg				17	
			1000			
					2	
	menu:coverage				1	
	menu:targets destroyed				1	
	characters typed into BDA				1	
	"Starttime"+typed characters				0	
	"Endtime"+typed characters				0	
	button:SEND BDA Report				0	
			1000			
	button:BDA				2	
	button:FreeText				17	
	button:MOVCMD				17	
	button:SPOT				1	
			1000			
	menu:addressees for MOVCMD Report				7	
	menu:tasks				7	
	menu:location				7	
	menu:When				17	
	menu:DTG				17	
	characters typed into MOVCMD				17	
	button:SEND MOVCMD Report				17	
			1000			
	menu:addressees for SPOT Report				3	
	menu:my activity				3	
	information:target				3	
	button:SEND SPOT Report				3	
			1000			
	scan pattern graphic				8	
	performance characteristics				8	

Results from Optimization Run on Comanche Data Script.3

Input Data		Results/Analysi
	1000	
search frame		18
button:BROWSE		6
button:AUTO		6
button:DELAY 5		6
	1000	
search frame		18
button:NO TGT		18
	1000	
map		15
button:LOS		15
	1000	
button:XMIT 1		1
button:XMIT 2		1
button:XMIT 3		1
button:XMIT 4		11
button:XMIT 5		11
button:XMIT PWR		11
	1000	
button:SQL 1		0
button:SQL 2		0
button:SQL 3		0
button:SQL 4		0
button:SQL 5		0
	1000	
Checklist:ADVS PROC		19
button:CHECK ADVS PROC		19
	1000	
Checklist:EMERG		5
button:CHECK EMERG PROC		5

Application of Optimizer to Display Optimizer User Interface
Input Files

`File:  all_fns.txt`

```
s Create Function Sequences
s Remove Function
s Remove Display Item
s Set Parameters
s Set Clusters
s Remove Clusters
s Remove Cluster Display Item
```

`File:  all_dis.txt`

```
All Functions list
Scenario Functions list
All Display Items list
Function Display Iems list
button: Create New Function
button: Remove Selected Function
button: Remove Selected disp it
button: Create new Display Item
button: OK (edit)
button: Cancel (edit)
button: Help (edit)
Criticality + entry box
Area + entry box
Create Home Page + entry box
Create Function Links+entry box
PageSize + entry box
# of Pages + entry box
LinkSize + entry box
button: OK (params)
button: Cancel (params)
Cluster list
Used Display Items list
Cluster Display Items list
Importance + entry box
button: Create New Cluster
button: Remove Selected Cluster
button: Remove Selected Cluster Display Item
button: OK (clust)
button: Cancel (clust)
button: Help (clust)
```

Page Organization Results for Display Optimizer User Interface

Page Size = 70

Page 3	**Total Area: 68.00**
All Functions list	Area: 8
Scenario Functions list	Area: 8
All Display Items list	Area: 8
Function Display Items list	Area: 8
button: Create New Function	Area: 2
button: Remove Selected Function	Area: 2
button: Remove Selected Display Items	Area: 2
button: Create New Display Items	Area: 2
Criticality + entry box	Area: 2
Area + entry box	Area: 2
Create Home Page + entry box	Area: 3
Create Function Links + entry box	Area: 3
PageSize + entry box	Area: 3
# of Pages + entry box	Area: 3
Link Size + entry box	Area: 3
button: OK (params)	Area: 2
button: Cancel (params)	Area: 2
Homepage	Area: 3
Link to Page 7	

Page 7	**Total Area: 40.00**
Cluster list	Area: 8
Used Display Items list	Area: 8
Cluster Display Items list	Area: 8
Importance + entry box	Area: 2
button: Create New Cluster	Area: 2
button: Remove Selected Cluster	Area: 2
button: Remove Selected Cluster Display Items	Area: 2
button: OK (clust)	Area: 2
button: Cancel (clust)	Area: 2
button: Help (clust)	Area: 2
Link to Page 3	

****

****

Button Press Analysis

1. Create Function Sequences

Pg: 3 BP: */* All Functions list (8)
Pg: 3 BP: 0/0 Scenario Functions list (8)
Pg: 3 BP: 0/0 All Display Items list (8)
Pg: 3 BP: 0/0 Function Display Items list (8)
Pg: 3 BP: 0/0 All Functions list (8)
Pg: 3 BP: 0/0 Criticality +entry box (2)
Pg: 3 BP: 0/0 Function Display Items list (8)
Pg: 3 BP: 0/0 Area + entry box (2)
Pg: 3 BP: 0/0 All Functions list (8)
Pg: 3 BP: 0/0 button: Create New Function (2)
Pg: 3 BP: 0/0 All Display Items list (8)
Pg: 3 BP: 0/0 button: Create New Display Item (2)

2. Remove Function

Pg: 3 BP: 0/0 Scenario Functions list (8)
Pg: 3 BP: 0/0 button: Remove Selected Functions (2)

3. Remove Display Item

Pg: 3 BP: 0/0 Function Display Items list (8)
Pg: 3 BP: 0/0 button: Remove Selected Display Items (2)

4. Set Clusters

Pg: 7 BP: 1/1 Cluster list (8)
Pg: 7 BP: 0/1 button: Create New Cluster (2)
Pg: 7 BP: 0/1 Used Display Items list (8)
Pg: 7 BP: 0/1 Cluster Display Items list (8)
Pg: 7 BP: 0/1 Importance + entry box (2)
Pg: 7 BP: 0/1 button: OK (clust) (2)
Pg: 7 BP: 0/1 button: Cancel (clust) (2)
Pg: 7 BP: 0/1 button: Help (clust) (2)

5. Remove Cluster

Pg: 7 BP: 0/1 Cluster list (8)
Pg: 7 BP: 0/1 button: Remove Selected Cluster (2)

6. Remove Cluster Display Item

Pg: 7 BP: 0/1 Cluster Display Items list (8)
Pg: 7 BP: 0/1 button: Remove Selected Cluster Display Items (2)

7. Set Parameters

Pg: 3 BP: 1/2 Create HomePage + entry box (3)
Pg: 3 BP: 0/2 Create Function Links + entry box (3)
Pg: 3 BP: 0/2 Page Size + entry box (3)
Pg: 3 BP: 0/2 # of Pages + entry box (3)
Pg: 3 BP: 0/2 Link Size + entry box (3)
Pg: 3 BP: 0/2 button: OK (params) (2)
Pg: 3 BP: 0/2 button: Cancel (params) (2)

****

Appendix F

Display Optimizer System Description

The Display Optimizer Prototype Software System

The Display Optimizer software system was developed in Phase I and demonstrated at the Phase I Final Presentation at NASA Ames Research Center to the COTR and others. This software system consists of three parts, the optimization algorithm and two user interface modules, one for entering input data and the other for presenting results of the optimization, as illustrated in the diagram in Figure F-1. The optimization algorithm, which forms the core of the Display Optimizer, was developed at Harvard University by Rebecca Hwa under the supervision of Dr. Shieber and Dr. Marks. This algorithm code was then embedded in a user-oriented, MS-Windows PC-based system developed at Tica Technologies, Inc., by James Kelly.

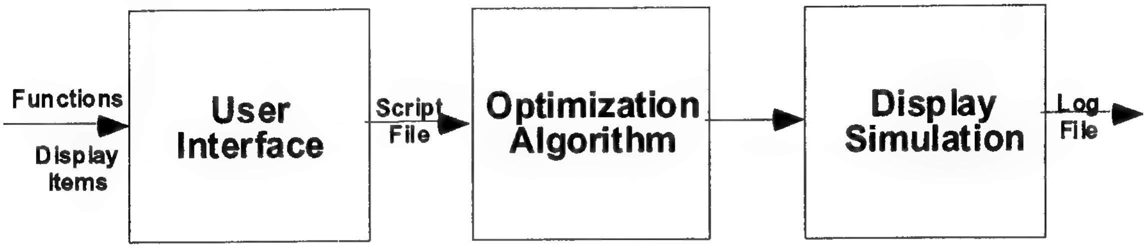


Figure F-1. Schematic Diagram of the Display Optimizer System

As shown in Figure F-1, the user provides two text files for use by the Display Optimizer. One file, named `all_fns.txt`, is a list of functions. The other file, named `all_dis.txt`, is a list of display items. These files provide lists of functions and display items for the user to manipulate to create a scenario on which the optimization of page organization will be based. Examples of these files for the Display Optimizer are shown in Appendix E. In the `all_fns.txt` file, the letters **s** and **e** preceding the function name are flags indicating whether the function is a normal function (**s**) or an emergency function (**e**). As originally designed, these files would provide initial lists to which the user could add functions and display items through a user interface dialog box. However, due to the limited resources of Phase I, the capability to create new functions and display items was not implemented. As the system is currently implemented at the end of Phase I, the user must provide all functions to appear in the scenario and all display items required by those functions in text file format. The creation of multiple scenarios may, however, be performed through the use of dialog boxes.

This Appendix contains figures illustrating the windows and dialog boxes which comprise the user interface for the Display Optimizer. The main application window is shown on pages F-6 and F-10. On page F-6, the File menu is visible, showing the options. Page F-10 shows the Run menu. By selecting New on the File menu, the user may create a new scenario. When the user clicks on the Save button, a standard Windows file save dialog box is displayed, through which the user may save the scenario he/she has created. When the user selects Open, a standard Windows Open File dialog box is displayed and the user may select a file containing a scenario that has previously been saved. After opening a saved scenario file, the user may click on the Edit button to edit the saved scenario. Clicking on Exit closes the application.

When the user clicks on New or Edit, the Edit Scenario dialog box is presented, as shown on page F-7. This dialog box contains four selectable list boxes. The first one on the left, labeled All Functions, presents the list of functions contained in the file `all_fns.txt`. The user builds his/her scenario by clicking on functions in the All Functions list in the order that they appear chronologically in the scenario. Clicking on a function in the All Functions list places that function on the Scenario Functions list, which constitutes the chronological list of functions which make up the scenario to be used as the basis for optimizing the assignment of display items to pages. As explained above, the optimization algorithm is based in part on the assumption that performance of each function requires access to certain display items in a particular sequence. So, to construct a scenario on which to base the optimization, the user must create the display item sequences for each function in the scenario. To do this, the user clicks on a function in the Scenario Functions list and then clicks on a display item in the All Display Items list. Clicking on a display item in the All Display Items list places that display item on the Function Display Items list, which constitutes the chronological sequence of display items required to perform the function selected in the Scenario Functions list.

Associated with each Scenario Function must be a Criticality, indicating the importance or criticality of that function relative to other functions in the scenario. The criticality value for the selected (highlighted) function is displayed in the box labeled Criticality and may be changed by typing in a new number. Similarly, the area required by the selected (highlighted) display item in the Function Display Items list is displayed in the box labeled Area and may be changed by typing in a new number. To completely specify a scenario, the user should assign criticality values to each Scenario Function and assign an area to each Function Display Item. A function can have only one criticality value and a display item can have only one area value. Once a criticality value is assigned to a function, it is associated with that function and need not be entered again if the function appears more than once in the scenario. The same holds true for the area assigned to a display item; it need be entered only once. As mentioned above, the ability to create new Functions and Display Items has not been implemented.

A function can be removed from the Scenario Functions list by selecting it and then clicking on the Remove Selection button located below the Scenario Functions list. Similarly, a display item can be removed from the Function Display Items list by selecting it and then clicking on the Remove Selection button located below the Function Display Items list.

The ability to specify *clusters* of display items is another way for the user to influence the design. The specification of clusters provides additional constraints to the optimization algorithm and allows the user to define conceptual clusters of display items that should be grouped on the same page if practicable. Clusters, being user-defined, may be unrelated to groupings determined by the display item sequences. The optimization algorithm will attempt to place cluster members on the same page, while attempting to preserve the other constraints that have been specified. The ability to define clusters allows the user to implement new constraints or to bias the design away from the sequential constraints obtained by the display item sequences.

Clusters are specified through an additional dialog box accessed by the Set Clusters button on the Edit Scenario window. The Edit Clusters dialog box is illustrated on page F-8. To create a cluster, the user clicks on the Create New Cluster button. Cluster 1 appears in the Clusters list box. When the user clicks on Cluster 1, it is highlighted; when the user clicks on a display item in the Display Items list box, that display item appears in the Cluster Display Items list. Clicking on additional display items places those display items on the Cluster Display Items list. A display item can be removed from the cluster by selecting it and then clicking on the Remove Selection button below the Cluster Display Items list. The user can remove a cluster in a similar way. It is recommended that before removing a cluster, the user first divest it of its display items. Each cluster is assigned a cluster Importance. When a cluster is selected and highlighted, its Importance is displayed in the box labeled Importance below the Clusters list. This value may be changed by typing a new number into this box. Clicking on OK makes the Edit clusters window disappear, bringing the user back to the Edit Scenario window.

In the Edit Scenario window there is another button labeled Set Parameters. Clicking on this button presents the Set Parameters window, illustrated on page F-9. Through this interface the user may enter a number of parameters affecting how the algorithm handles the optimization. The first parameter is a flag indicating whether to Create a Home Page or not. If yes is shown in the box, the interface module will create scripts which provide for a page to be designated as a home page which can be accessed directly from any page and from which there will be easy access to the starting page for each function. If anything other than yes appears in that box, this feature will not be provided. In that case, the algorithm will minimize access responses within function sequences and within clusters, but will not attempt to create a home page access structure.

The second parameter is a flag indicating whether to Create Function Links or not. If yes is shown in the box, the interface module will create scripts which are essentially display item sequences linking the last display item in a function with the first display item in the function which immediately follows it in the chronological list of Scenario Functions. This provision may in some cases result in better accessibility to display items in the order they are required in the scenario. When anything other than yes appears in the Create Function Links box, this feature will not be provided and the sequential relationships between functions in the scenario will be ignored.

The next parameter, Page Size, refers to the total area available on the display screen. This value is obviously related to the area values entered for each display item. It is the user's choice as to the meaning of these numbers, but total page size and area required by each display item must be on the same scale. In our demonstration we chose values that could be interpreted as percentages. Alternatively, one could choose numbers that could be interpreted as square inches, or some other measure. In any case, it must be remembered that this algorithm is concerned only with allocation of display items to pages, not with layout. The algorithm does not take two-dimensional packing constraints into account. It merely treats these area or size parameters as one dimensional values.

The parameter # of Pages is a technical parameter that scales with the size of the problem. This value is the number of pages that the algorithm begins with in the process of minimizing the access cost. Ultimately, the algorithm should have its own algorithm for setting this value, but in the experimental stages of development, this parameter has been made accessible to the user. A designer/user of a more fully developed system would not have to deal with such a technical parameter.

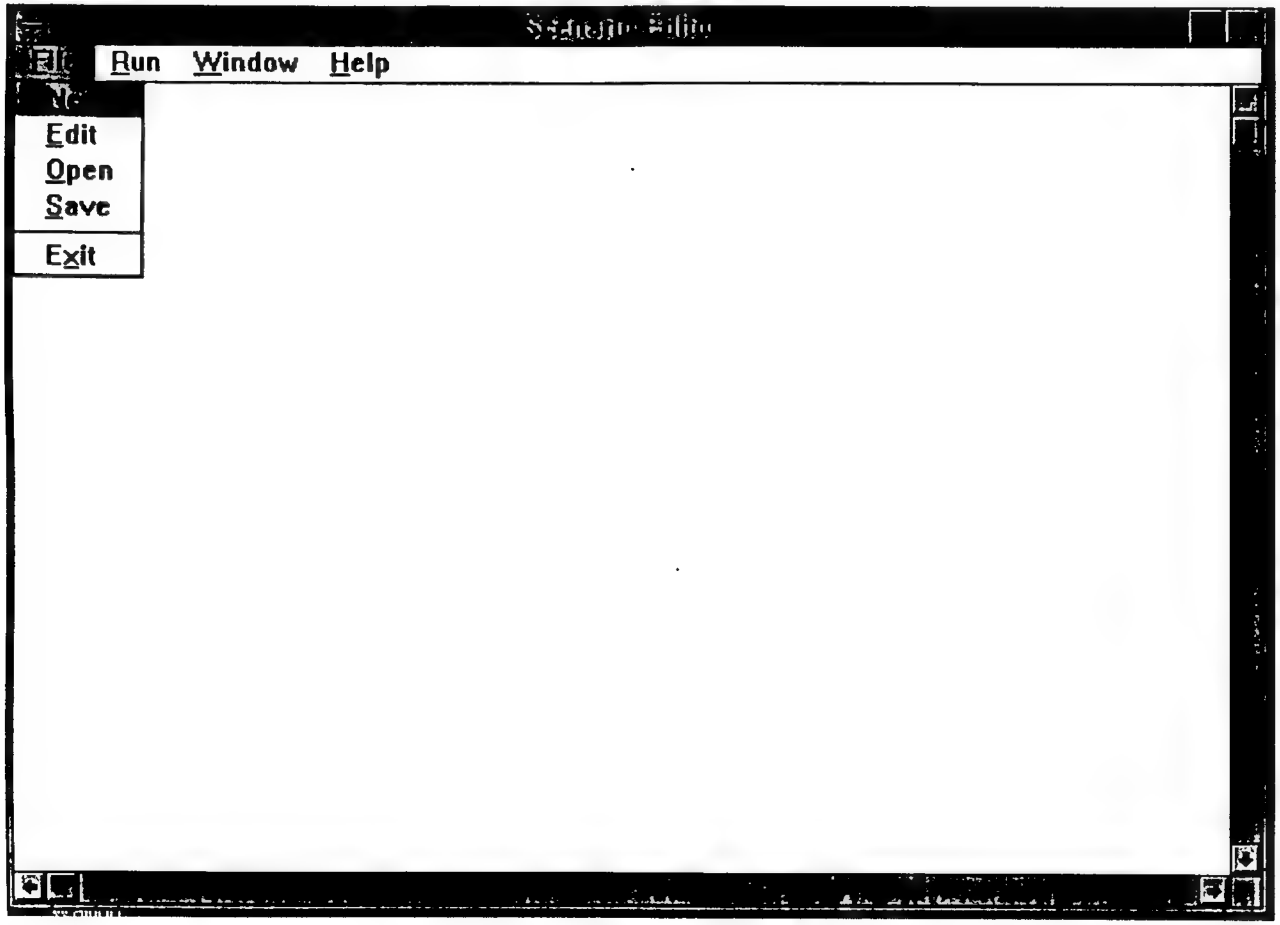
The last parameter, Link Size, is the area assigned to link buttons which provide a means of accessing one page from another. This parameter should be thought of in the context of the other display items specified, for it specifies the size of a button that would appear on a page along with other display items. An access button can be thought of as another display item, but one created as needed by the page organization optimizer. In the context of the other display items, the button size should be consistent with other buttons which may have been specified as display items. Clicking on OK makes the Set Parameters window disappear, saving the values which appear in the boxes, and returning the user to the Edit Scenario window.

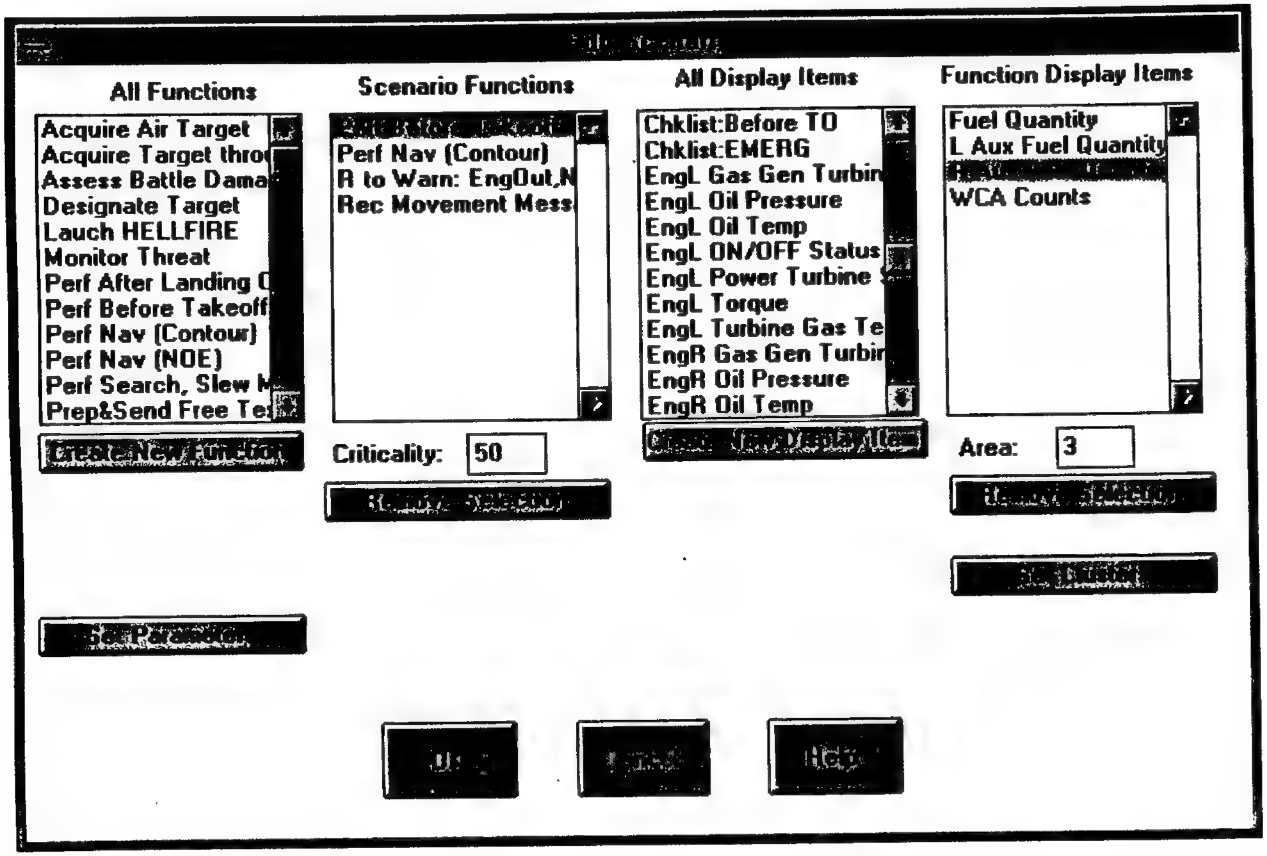
In the Edit Scenario window, clicking on OK makes the window disappear and returns the user to the main application window with the main menu bar. To preserve the results of the scenario editing session, the user must click on the Save button on the File menu. If desired, the user may save the scenario to a file. Exiting the Save dialog box by saving a file (or by clicking on Cancel) also results in saving a file named `script`, which provides the results of the scenario editing session to the algorithm code. The user may now run the optimization by clicking on the Optimization button on the Run menu. (See page F-10 for an illustration of this menu.)

When the optimization has finished, two types of output are printed to the screen. The first is a listing of the contents of each page that results from the optimization. An example of such a listing is shown on page F-11. For each page, the total area filled is given followed by a list of the display items that are assigned to that page, with the area for each, and then the links to other pages created by the algorithm. Because this listing is rather difficult to relate to the scenario that was created by the user and used by the optimization algorithm, another way of representing the optimization results is also available. An example of this type of output is shown on page F-12. This output format includes an analysis we have called the Button Press Analysis since it consists of a chronological list of scenario functions with the display item sequences for each function, together with the page to which each display item has been assigned. This essentially results in a sequence of display items to be accessed throughout the scenario and a chronological listing of pages that must be accessed in order. From this we generate a tally of button presses. An example is "BP 1/4," where the first number is the number of button presses to get from the previous page to the current page and the second number is the total number of button presses since the most recent *. (An * means that there is no direct link from the previous page to the current page. The total tally is restarted from each *. An * may mean that more than one button press is required to access the current page from the previous page, or it may, in the worst of cases, mean that there is no access possible. For the Phase I prototype, no attempt was made to do more sophisticated analysis in the case where direct access from the previous page to the current page is not possible.) The total number of button presses might be used by the designer/user to evaluate the result of changing certain parameters, such as display item areas or total page area, or perhaps different clustering. Included in this output format are the areas for each display.

In addition to the screen printouts described above, we provided another way of visualizing the optimization results. In simulation mode, the system simulates the performance of the scenario functions in the context of the page organization resulting from the optimization. With limited Phase I resources, this simulation capability is still very crude. A much better framework for creating such a simulation could be provided by MIDAS or another simulation tool. In the case of our Phase I prototype, the user may click on the Simulation button in the Run menu to see a crude window representing the display page with navigation buttons. The Display Simulation window is illustrated on page F-13. For each page, the page number is shown in the upper left corner and the display items that have been assigned to that page are listed in the center panel (remember that our Phase I effort involves no page layout, only page organization). When the algorithm has established a link from the current page to another page, a label is shown next to one of the buttons in the Display Simulation window. The example on page F-13 shows three links, to pages 1, 2 and 3. Clicking on the button next to the To Page 1 label will present Page 1 with its display items and links. By referring to the scenario and its list of functions and their display item sequences the designer/user may step through the scenario making note of the efficiency of access to the required information. This output format does not actually contain any more information than the previous screen printout, but it gives a much greater understanding of access costs and cluster effectiveness than the printout.

The Display Optimizer writes out two files, one for each of its presentation modes. The first, called `log.txt`, replicates the screen display of page organization and the button press analysis. The other, called `A1`, is a more detailed record of progress through the optimization process, including the initial random page allocation from which the algorithm begins its optimization process.


Scenario Editor
Run
Window
Help
Edit
Open
Save
Exit



All Functions

- Acquire Air Target
- Acquire Target thro
- Assess Battle Dama
- Designate Target
- Lauch HELLFIRE
- Monitor Threat
- Perf After Landing C
- Perf Before Takeoff
- Perf Nav (Contour)
- Perf Nav (NOE)
- Perf Search, Slew M
- Prep&Send Free Te

Scenario Functions

- Perf Nav (Contour)
- R to Warn: EngOut,N
- Rec Movement Mess

Criticality: 50

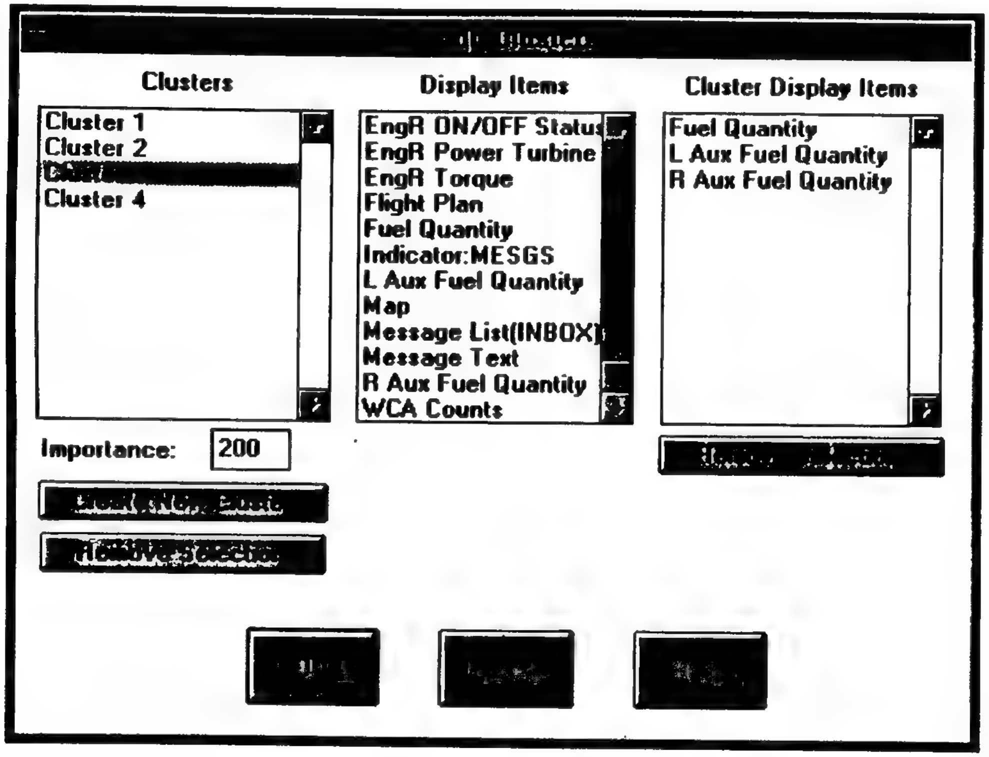
All Display Items

- Chklist:Before TO
- Chklist:EMERG
- EngL Gas Gen Turbin
- EngL Oil Pressure
- EngL Oil Temp
- EngL ON/OFF Status
- EngL Power Turbine
- EngL Torque
- EngL Turbine Gas Te
- EngR Gas Gen Turbir
- EngR Oil Pressure
- EngR Oil Temp

Function Display Items

- Fuel Quantity
- L Aux Fuel Quantity
- WCA Counts

Area: 3



Clusters:

Cluster 1
Cluster 2
Cluster 3
Cluster 4

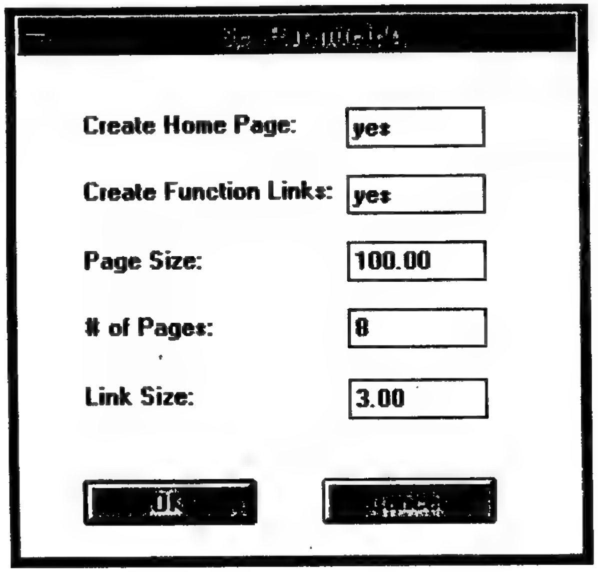
Display Items:

EngR ON/OFF Status
EngR Power Turbine
EngR Torque
Flight Plan
Fuel Quantity
Indicator:MESGS
L Aux Fuel Quantity
Map
Message List(INBOX)
Message Text
R Aux Fuel Quantity
WCA Counts

Cluster Display Items:

Fuel Quantity
L Aux Fuel Quantity
R Aux Fuel Quantity

Importance: 200



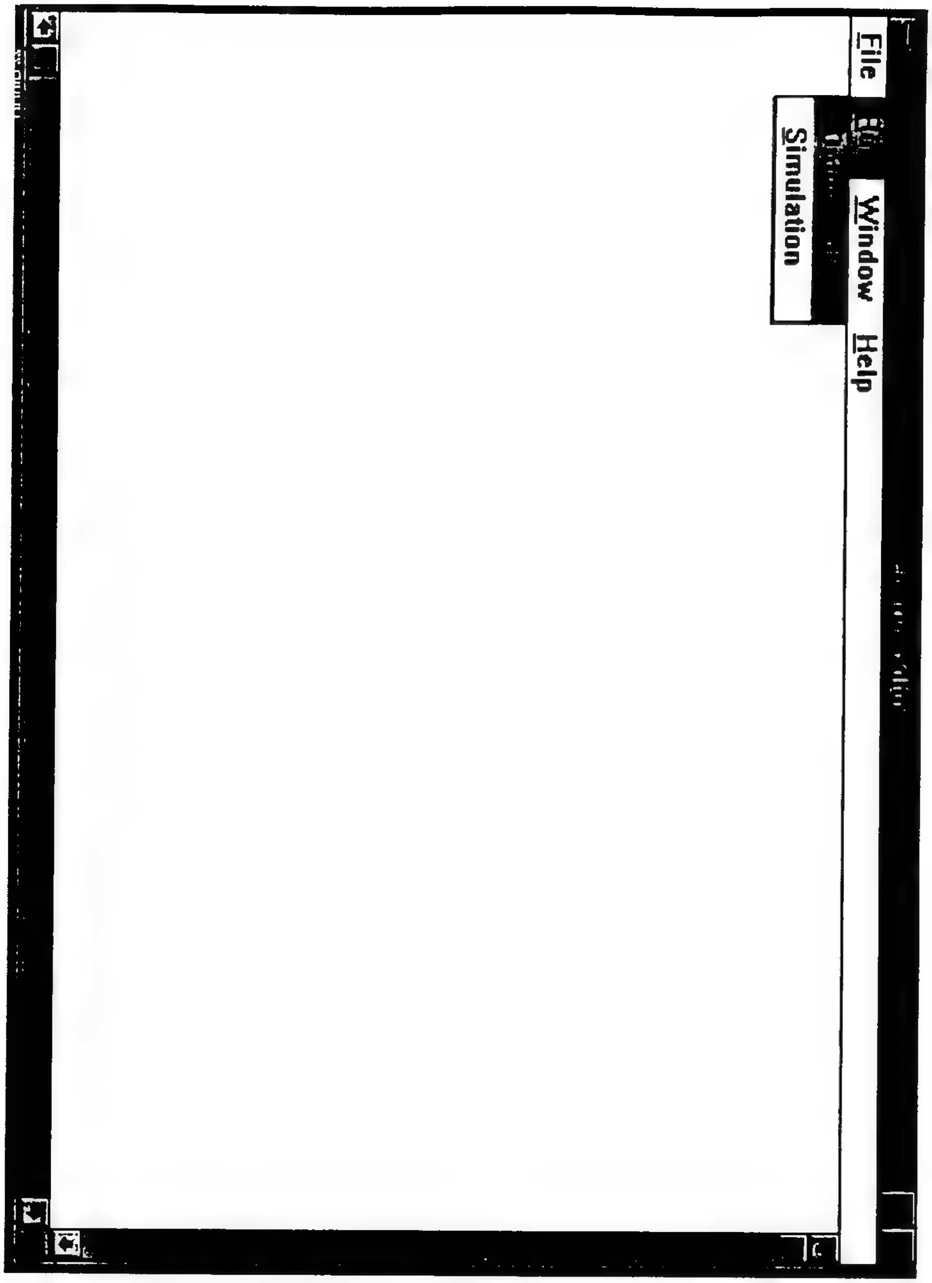
Create Home Page: yes

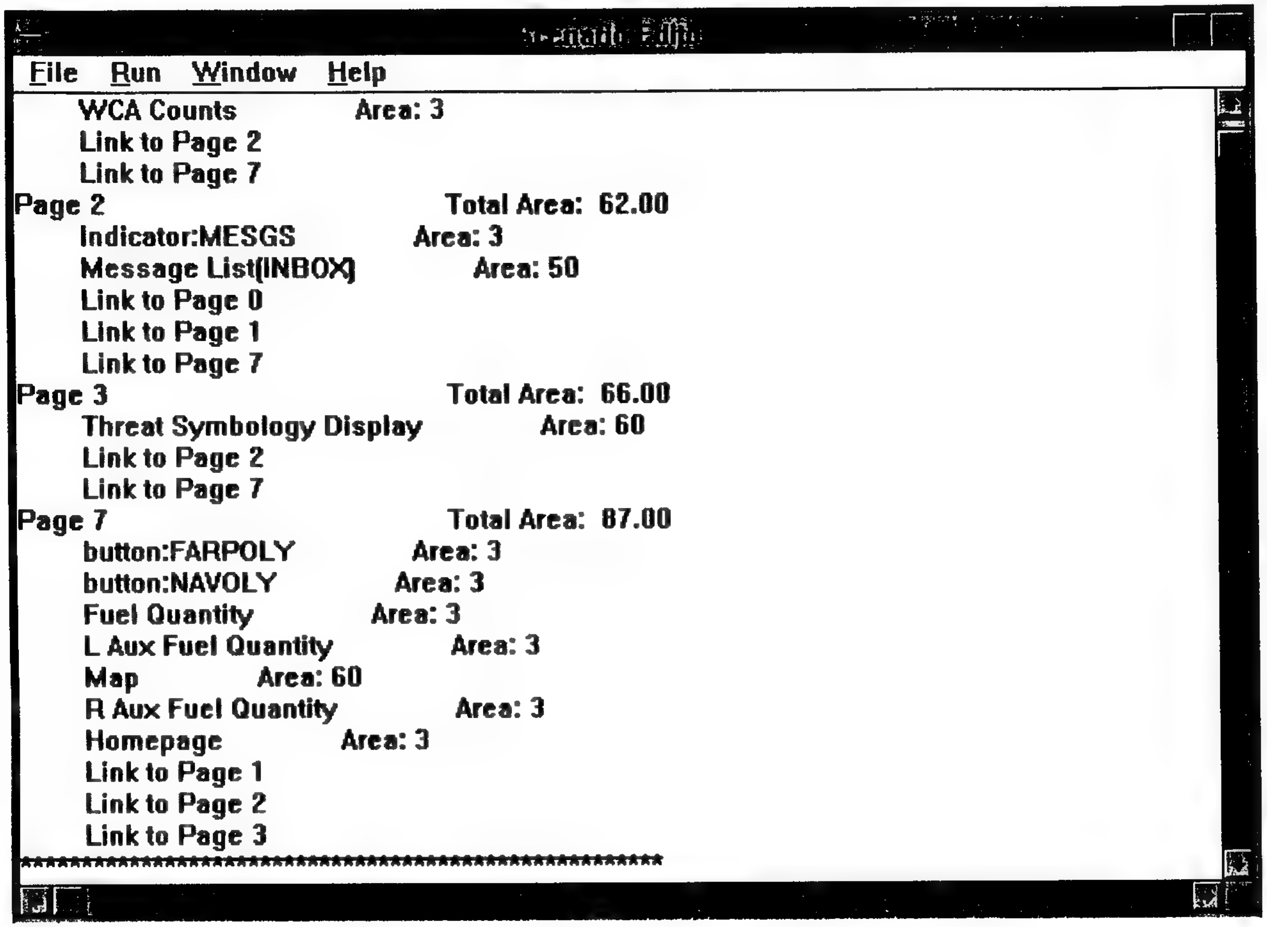
Create Function Links: yes

Page Size: 100.00

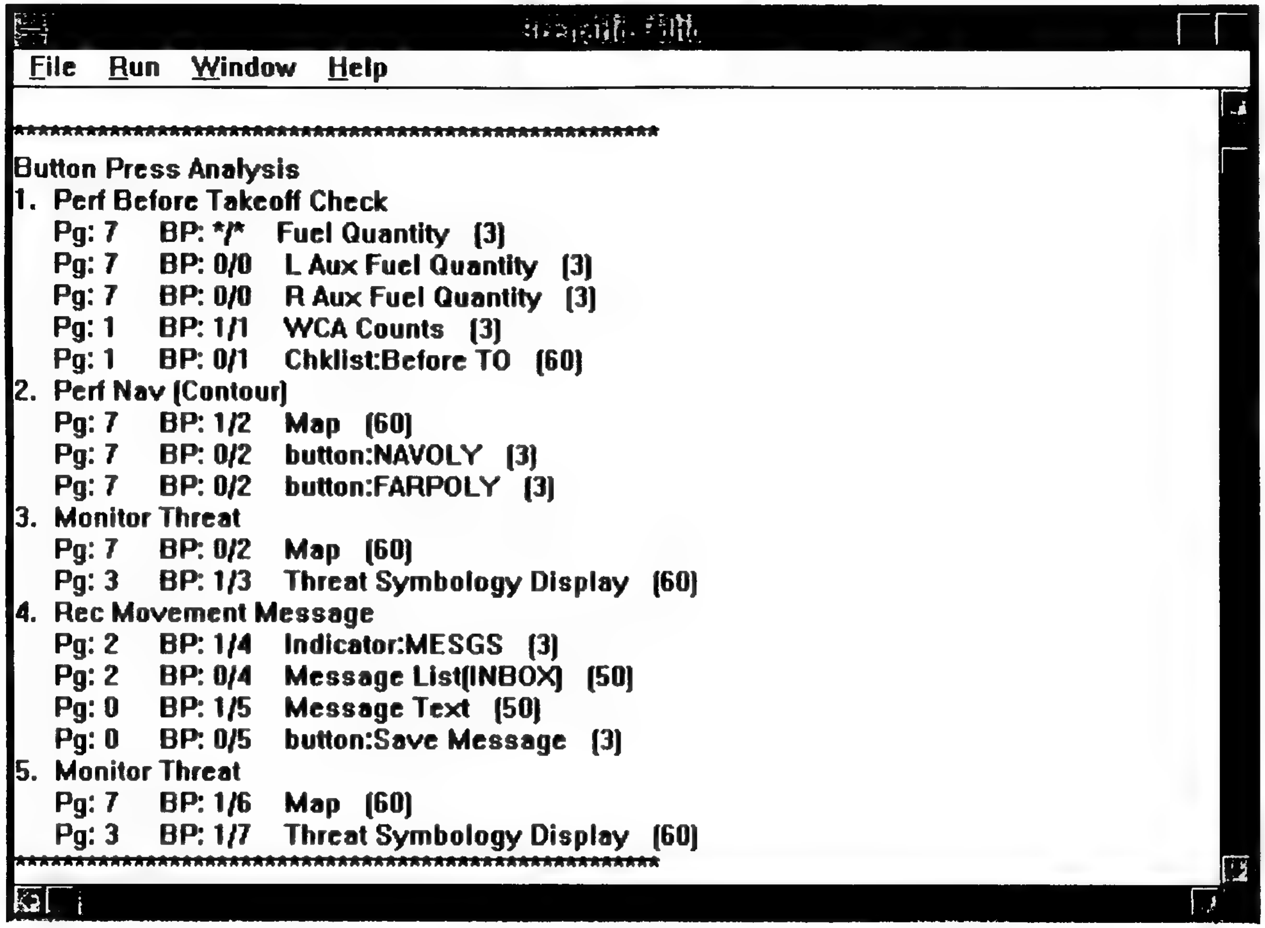
# of Pages: 8

Link Size: 3.00

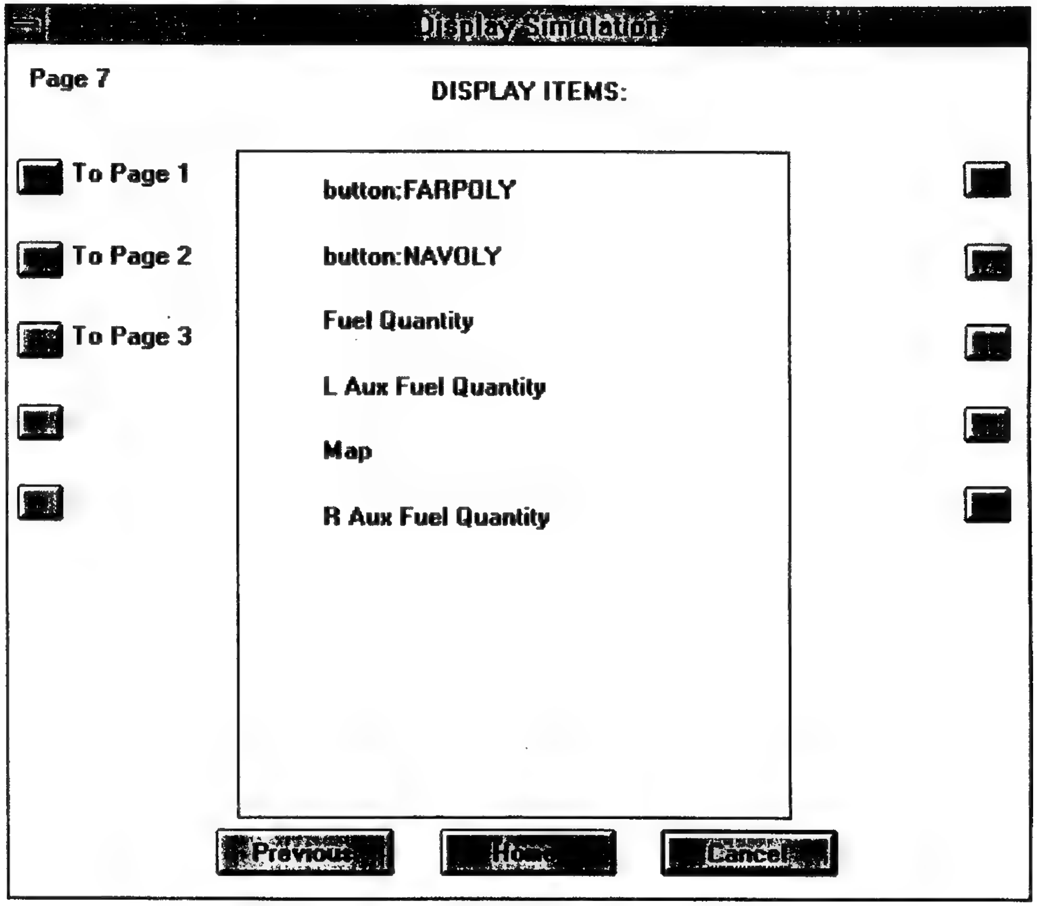

File
Window
Help
Simulation



```
     WCA Counts          Area: 3
     Link to Page 2
     Link to Page 7
Page 2                         Total Area:  62.00
     Indicator:MESGS          Area: 3
     Message List(INBOX)           Area: 50
     Link to Page 0
     Link to Page 1
     Link to Page 7
Page 3                         Total Area:  66.00
     Threat Symbology Display           Area: 60
     Link to Page 2
     Link to Page 7
Page 7                         Total Area:  87.00
     button:FARPOLY          Area: 3
     button:NAVOLY          Area: 3
     Fuel Quantity          Area: 3
     L Aux Fuel Quantity           Area: 3
     Map          Area: 60
     R Aux Fuel Quantity           Area: 3
     Homepage          Area: 3
     Link to Page 1
     Link to Page 2
     Link to Page 3
**********************************************************************
```



```
*******************************************************
Button Press Analysis
1.  Perf Before Takeoff Check
    Pg: 7    BP: */*   Fuel Quantity  [3]
    Pg: 7    BP: 0/0   L Aux Fuel Quantity  [3]
    Pg: 7    BP: 0/0   R Aux Fuel Quantity  [3]
    Pg: 1    BP: 1/1   WCA Counts  [3]
    Pg: 1    BP: 0/1   Chklist:Before TO  [60]
2.  Perf Nav [Contour]
    Pg: 7    BP: 1/2   Map  [60]
    Pg: 7    BP: 0/2   button:NAVOLY  [3]
    Pg: 7    BP: 0/2   button:FARPOLY  [3]
3.  Monitor Threat
    Pg: 7    BP: 0/2   Map  [60]
    Pg: 3    BP: 1/3   Threat Symbology Display  [60]
4.  Rec Movement Message
    Pg: 2    BP: 1/4   Indicator:MESGS  [3]
    Pg: 2    BP: 0/4   Message List[INBOX]  [50]
    Pg: 0    BP: 1/5   Message Text  [50]
    Pg: 0    BP: 0/5   button:Save Message  [3]
5.  Monitor Threat
    Pg: 7    BP: 1/6   Map  [60]
    Pg: 3    BP: 1/7   Threat Symbology Display  [60]
*******************************************************
```



Page 7

DISPLAY ITEMS:

button:FARPOLY

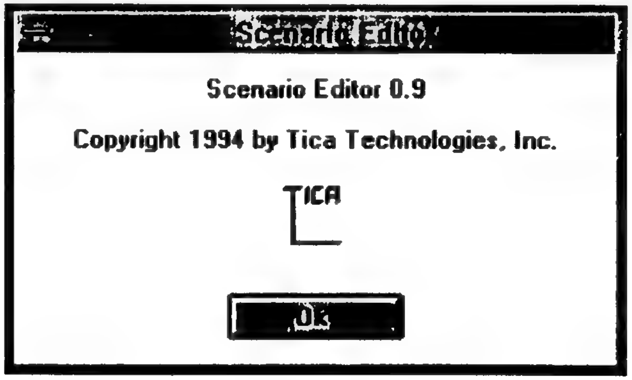
button:NAVOLY

Fuel Quantity

L Aux Fuel Quantity

Map

R Aux Fuel Quantity



Scenario Editor 0.9

Copyright 1994 by Tica Technologies, Inc.

Appendix G

A Genetic Algorithm for Graph Partitioning

MITSUBISHI ELECTRIC RESEARCH LABORATORIES
CAMBRIDGE RESEARCH CENTER

A Stochastic Search Technique for Graph Bisection

Joe Marks
MERL

Stuart M. Shieber
Harvard University

J. Thomas Ngo
Interval Research Corp.

TR-94-18 November 1994

Abstract

We present a new heuristic algorithm for graph bisection. This heuristic combines stochastic search and an implicit notion of clustering in a novel manner. In comparison with a large-sample, time-equated set of runs of the Kernighan-Lin algorithm, the new algorithm demonstrates a modest but significant superiority in terms of the best bisections found.

1 Introduction

Given a graph $G = (V, E)$ with an even number of vertices, the graph-bisection problem is to divide V into two equal-sized subsets X and Y such that the number of edges connecting vertices in X to vertices in Y (the size of the *cut set*, notated $\text{cut}(X, Y)$) is minimized. This problem is NP-complete [6]. Graph bisection and its generalizations[1] have considerable practical significance, especially in the areas of VLSI design and operations research.

The benchmark algorithm for graph bisection is due to Kernighan and Lin [11]. (The efficient implementation of this heuristic technique was described by Fiduccia and Mattheyses [4], so the algorithm is sometimes referred to as the Kernighan-Lin-Fiduccia-Mattheyses algorithm.) The Kernighan-Lin (KL) algorithm improves an initial random bisection by making a sequence of locally optimal vertex swaps between the subsets X and Y. The vertex-swap operation is also the primitive perturbation operator used in applications of simulated annealing to graph bisection [12, 13]. In spite of the folk wisdom that simulated annealing is capable of avoiding the local minima that often plague greedy heuristics like the KL algorithm, Johnson et al. [10] found that the relative performance of the two algorithms depends on the nature of the graphs being bisected: simulated annealing has an advantage on sparse, relatively uniform graphs, but KL is better for graphs with structure.[2]

Recently, more aggressive attempts have been made to exploit the structure that is often found in graphs of practical significance. The common theme of these attempts is clustering: by grouping together vertices in tightly connected subgraphs, clusters of vertices can be treated as individual supernodes during the application of standard heuristics like KL or simulated annealing. The various incarnations of the clustering idea appear to show a marked superiority over the original KL algorithm [1, 2, 3, 5, 8, 9, 14, 15, 16], though the degree of superiority is unclear because the reported empirical results tend to sell the KL algorithm short, as we will argue below.

The algorithm we describe in this paper can be considered a synthesis of ideas from previous work: it includes a very simple implicit clustering heuristic, employs a stochastic search strategy (like simulated annealing or a genetic algorithm [7]), and uses the KL algorithm for final refinement of the computed bisections. When compared fairly with the KL algorithm (i.e., giving each algorithm equal time and ensuring that a large sample of KL runs is considered), the new algorithm exhibits significant superiority on a variety of test graphs.

In the following sections we describe the algorithm, present an empirical analysis of its behavior, and conclude with a discussion of future work.

[1] More general classes of graph-partitioning problems arise when V can be divided into more than two subsets, when the strict equality constraint on the sizes of the subsets is relaxed, and when weights are associated with the vertices and edges to be used in the constraint-satisfaction and cut-set-size computations.

[2] The conclusions that Johnson and his colleagues drew from their thorough empirical analysis are more complicated and informative than this simple précis suggests, but the statement is approximately true.

2 Algorithm Description

Our algorithm is based on a simple *seed-growth* heuristic.[3] We start with two disjoint, equal-sized subsets of the vertex set to seed the two partitions, and add the remaining vertices one at a time into alternate partitions, at each step choosing the vertex to be added in a greedy manner. When adding to partition X we choose a vertex a that minimizes $\mathrm{cut}(\{a\}, Y) - \mathrm{cut}(\{a\}, X)$; intuitively, we minimize the number of edges added to the cut set separating X and Y while maximizing the number of edges barred from future addition to the cut set. Thus the notion of clustering is implicit in this heuristic, as compared to heuristics in which explicit clusters are computed and manipulated [1, 2, 3, 5, 8, 9, 14, 15, 16].

More formally, the algorithm can be given by the following pseudocode. (All underlined quantities are parameters of the heuristic that can be varied. The values given in the paper are those that gave the best empirical results in an initial set of experiments.)

```
Input: An undirected graph $G = (V, E)$. $|V|$ is assumed to be even.

Output: A partition of $V$ into subsets $X$ and $Y$ of size $\frac{|V|}{2}$.

Procedure:

    1. Let the seed sets $s_x$ and $s_y$ be randomly chosen disjoint subsets of $V$ such that
       $|s_x| = |s_y| = \lfloor \underline{0.01}\,|V| \rfloor$.
    2. $X \leftarrow s_x; Y \leftarrow s_y$.
    3. Repeat substeps (a) and (b) until all the vertices in $V$ have been assigned to $X$
       or $Y$:
        (a) Find an unassigned vertex $a \in V$ such that $\mathrm{cut}(\{a\}, Y) - \mathrm{cut}(\{a\}, X)$ is
            minimal.
            $X \leftarrow X \cup \{a\}$.
        (b) Find an unassigned vertex $b \in V$ such that $\mathrm{cut}(\{b\}, X) - \mathrm{cut}(\{b\}, Y)$ is minimal.
            $Y \leftarrow Y \cup \{b\}$.
```

One application of the seed-growth heuristic is not likely to be particularly useful (on average it will be worse than a single application of the KL algorithm), but the $O(|V| + |E|)$ seed-growth heuristic—which is roughly an order of magnitude faster than an efficient implementation of the KL algorithm on standard test graphs—can be rendered effective by running it many times as part of a general search procedure. One such search procedure, a form of parallel hill climbing, is given here, though others (e.g., simulated annealing and genetic algorithms) might also be used effectively in combination with the seed-growth heuristic. The KL algorithm is used as a postprocess to achieve final refinement of the best bisections found by the search procedure.

[3] This heuristic bears some resemblance to the epitaxial-growth heuristic of Donath [3].

Input: An undirected graph $G = (V, E)$.

Output: A partition of V into subsets X and Y of size $\frac{|V|}{2}$.

Procedure:

1. Randomly choose a set P of $\underline{100}$ pairs (s_x, s_y) of seed sets using Step 1 of the seed-growth heuristic.
2. Compute the corresponding bisection (X, Y) for each seed-set pair $(s_x, s_y) \in P$ using Steps 2 and 3 of the seed-growth heuristic.
3. Repeat substeps (a) through (e) $\underline{5,000}$ times:
   (a) Randomly pick a seed-set pair $(s_x, s_y) \in P$.
   (b) Randomly select a vertex in one of s_x or s_y and replace it with another randomly chosen seed vertex from $V - s_x \cup s_y$; call the resulting seed-set pair (s'_x, s'_y).
   (c) Compute the corresponding bisection (X', Y') using Steps 2 and 3 of the seed-growth heuristic.
   (d) If $\text{cut}(X', Y') < \text{cut}(X, Y)$ then replace (s_x, s_y) in P with (s'_x, s'_y).
   (e) Every $\underline{1,000}$th iteration perform the following steps:
      i. Use the cut-set sizes of the corresponding bisections (i.e., the values of $\text{cut}(X, Y)$) to rank order the seed-set pairs (s_x, s_y) in P.
      ii. Replace the bottom $\underline{50}$ seed-set pairs in P with copies of the top $\underline{50}$ seed-set pairs in P.
4. Use the cut-set sizes of the corresponding bisections to rank order the seed-set pairs (s_x, s_y) in P.
5. For the top $\underline{20\%}$ of seed-set pairs (s_x, s_y) in P apply the KL algorithm to (X, Y); return the best bisection found.

Because this algorithm combines parallel hill climbing (PHC), the seed-growth (SG) heuristic, and the KL algorithm, we will refer to it as PHC/SG+KL.

3 Empirical Analysis

Heuristic algorithms for graph partitioning like the one described here cannot be evaluated in a purely analytic fashion; empirical analysis is the only way to ascertain such an algorithm's utility. Unfortunately, empirical analysis of algorithm performance is often done poorly, which sometimes leads to erroneous conclusions. In the following subsection we discuss two common errors that are often committed in the empirical analysis of graph-partitioning algorithms. We then present empirical results for our algorithm.

	KL: 20 runs		X: 20 runs		% improvement over KL	
Graph	min	avg	min	avg	min	avg
19ks	1131	1701.90	1154	1391.40	-2.03	18.24
5655	633	866.90	608	698.70	3.95	19.40
8870	70	118.15	69	95.10	1.43	19.51
PrimGA1	312	384.10	293	345.65	6.09	10.01
PrimGA2	1262	1716.30	915	1405.50	27.50	18.11
Test02	1177	1296.60	1195	1242.10	-1.53	4.20
Test03	906	2590.30	843	1503.60	6.95	41.95
Test04	1216	1316.35	1201	1245.55	1.23	5.38
Test05	2119	4524.55	1866	2113.95	11.94	53.28
Test06	1203	1580.10	1192	1285.95	0.91	18.62
bm1	302	385.10	229	327.80	24.17	14.88

Table 1: Kernighan-Lin and Algorithm X: an empirical comparison. Algorithm X runs five times more slowly than the Kernighan-Lin (KL) algorithm.

3.1 Caveats

Consider the evidence presented in Table 1. (This example is based on an empirical analysis reported by Wei and Cheng [16].) The table contains the average and minimum cut-set sizes of 11 graph bisections, computed from 20 runs of the KL algorithm and 20 runs of Algorithm X.[4] Although Algorithm X is five times more expensive than the KL algorithm, one might be tempted to conclude that the extra expense is indeed worthwhile, because its performance appears to be significantly better. However, the difference in performance is due solely to the extra time afforded Algorithm X, because Algorithm X merely returns the best of five runs of the KL algorithm! The moral is clear: Given the high variance of the distribution of results generated by the KL algorithm, any analysis that does not give equal time to KL will result in an inappropriate comparison.

The nature of the distribution of KL results provides a further opportunity for misleading analysis. Figure 1 shows the distribution of 10,000 values returned by the KL algorithm for graph bm1, which is derived from a circuit in the standard MCNC test suite. Suppose that Algorithm Y also generates a distribution of results with better mean but smaller variance: for instance, let us assume that it essentially always finds a bisection with cut-set size between 250 and 300 for this graph. If one compares the best result from m runs of Algorithm Y with the best result from n runs of the KL algorithm to determine which algorithm is better (where m and n have been chosen to equate overall running times, of course), the answer one gets will be affected by the magnitude of n. By inspection, roughly 1% of the values in the histogram for KL are less than 250. A simple probabilistic analysis shows that n

[4] The graphs were derived from circuit hypergraphs that were made available for the Microelectronics Center of North Carolina (MCNC) Layout Synthesis Workshop.

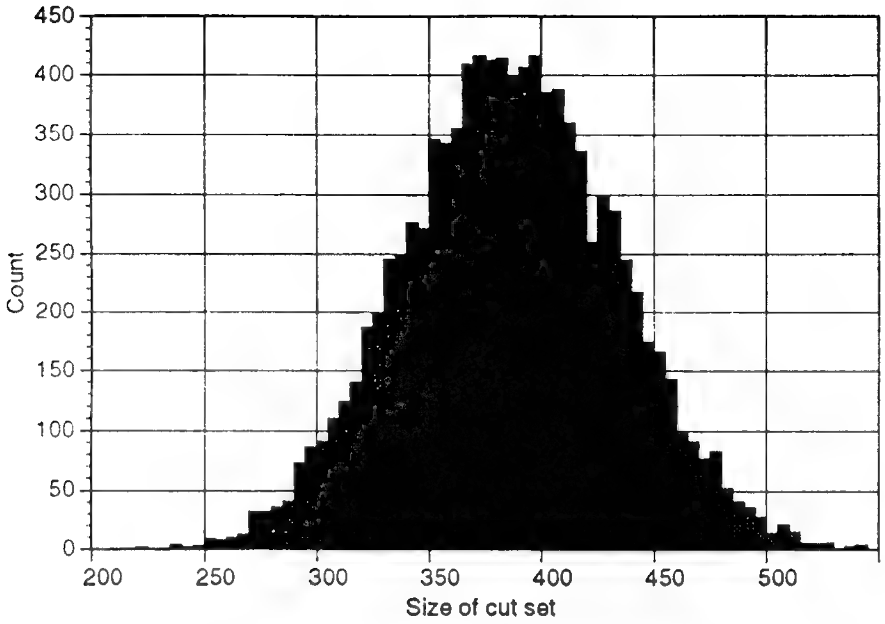


Figure 1: Histogram of values computed by the KL algorithm for graph bm1.

must be around 690 in order for KL to have at least a 50% chance of being declared the better algorithm by virtue of finding the best bisection. Therefore, if one can wait the hour or so required for 1000 runs of KL—as is typical for most applications involving graph partitioning—KL should be considered the better algorithm on the basis of this empirical evidence: it will very likely find a bisection with a smaller cut set than Algorithm Y. When absolute performance is what matters most, several tens or even hundreds of runs of the KL algorithm may be required to do it justice; a statistical analysis of the distribution of results for a given graph can be used to estimate an appropriate minimum number of runs, if such an estimate is needed [15]. Conversely, any comparisons with KL that involve as few as 10 or 20 runs—especially against algorithms with good average performance but low variance—would appear to be suspect, though such comparisons are not uncommon [2, 9, 16, 17].

3.2 Results

Table 2 contains an empirical comparison of the KL and PHC/SG+KL algorithms. The algorithms were tested on 13 graphs, 11 of which were derived from hypergraphs in the MCNC test suite, and two of which have been used for empirical testing in the operations-research community.[5]

[5] These graphs are instances of $G(1000, 0.0025)$ and $U(1000, 0.04)$. Graphs in $G(n,p)$ have n vertices, and the probability that there is an edge between any given pair of vertices is p. Graphs in $U(n,d)$ have n vertices that are randomly distributed on a unit square, and an edge exists between any pair of vertices that are distance d or less apart. One would expect the graphs in U, but not the graphs in G, to have exploitable structure [10].

					KL		PHC/SG+KL		
		avg	Time	# of	avg min		avg min		% impr.
Graph	$\lvert V\rvert$	deg	(secs)	runs	cut-set size	σ	cut-set size	σ	over KL
19ks	2844	93.2	12368	512	1020.8	33.5	976.8	89.2	4.3%
5655	922	20.1	1289	702	603.2	4.3	595.4	0.5	1.3%
8870	502	9.7	377	728	52.8	1.3	52.0	0.0	1.5%
PrimGA1	834	11.3	1054	628	235.6	15.6	218.8	0.8	7.1%
PrimGA2	3014	18.0	13785	420	1051.6	77.5	574.6	31.2	45.4%
Test02	1664	100.1	4245	486	1172.8	11.9	1164.2	22.1	0.7%
Test03	1608	71.2	3981	608	821.8	8.1	804.4	0.5	2.1%
Test04	1516	137.1	3589	454	1191.2	2.2	1184.0	3.1	0.6%
Test05	2596	167.3	10409	420	1887.4	26.4	1813.0	2.4	3.9%
Test06	1752	114.7	4718	500	1194.4	3.6	1188.4	2.3	0.5%
bm1	882	10.7	1176	570	240.4	14.5	209.2	1.3	13.0%
$G(1000, 0.0025)$	1000	2.5	1538	234	98.2	3.0	93.8	1.5	4.5%
$U(1000, 0.04)$	1000	5.0	1507	380	28.6	5.7	4.2	0.4	85.3%

Table 2: Kernighan-Lin and PHC/SG+KL: an empirical comparison.

For each graph in the test suite the following data are presented:

1. *Graph cardinality:* The number of vertices in the graph ($|V|$).

2. *Average degree:* The average number of edges incident upon a vertex in the graph.

3. *Running time:* The running time, in seconds, of the PHC/SG+KL algorithm on a Hewlett-Packard 735 workstation. (The running times range from a little under four hours for graph PrimGA2 to a little over six minutes for graph 8870.)

4. *Number of KL runs:* The number of runs of the KL algorithm that will take an amount of time equivalent to that required for the PHC/SG+KL algorithm.

5. *Average minimum cut-set size for KL:* The average minimum cut-set size found over five tests of k runs each, where k is the number of runs required for time equivalence with the PHC/SG+KL algorithm.

6. *Standard deviation of minimum cut-set size for KL:* The standard deviation of the minimum cut-set size found over the five tests.

7. *Average minimum cut-set size for PHC/SG+KL:* The average minimum cut-set size found over five runs of the PHC/SG+KL algorithm.

8. *Standard deviation of minimum cut-set size for PHC/SG+KL:* The standard deviation of the minimum cut-set size found over the five tests.

9. *Improvement over KL:* The average improvement of the PHC/SG+KL algorithm over the KL algorithm, expressed as a percentage of the average minimum cut-set size for KL.

In all cases, PHC/SG+KL generates better solutions than the large-sample, time-equated tests of KL. The advantage ranges from less than 1% to over 85%.

The results for PHC/SG+KL may appear modest relative to the results that have been reported recently for various clustering heuristics.[6] However, this is due in large part to the better results we report for KL because of the large number of KL runs we use, on average about 500. Recall that Table 1 shows the improvement one can get by taking the best of 100 runs of the KL algorithm versus the best of 20 runs; moreover, the best of 500 runs is quite an improvement, on average, on the best of 100 runs. Thus, our results cannot be directly compared to those previously published. We hope to replicate the results on other algorithms in the near future so as to allow comparison of PHC/SG+KL with other methods.

An interesting aspect of the data is the variation in relative performance of the algorithms: although PHC/SG+KL is superior to KL across the board, the degree of superiority differs markedly. For some graphs (5655, 8870, Test02, Test03, Test04 and Test06) the improvement is very small; for others (19ks, PrimGA1, Test05 and $G(1000, 0.0025)$) the improvement is small, but significant; and for the remaining three graphs (PrimGA2, bm1, and $U(1000, 0.04)$) the improvement is substantial. There is no obvious correlation between the degree of relative superiority of the PHC/SG+KL algorithm and the cardinality or average degree of the graphs in question.

For hybrid algorithms that involve the KL algorithm, the following question naturally arises: How much work is the KL part doing? In Table 3, an approximately time-equated comparison of the KL and PHC/SG algorithms is presented. (PHC/SG is the PHC/SG+KL algorithm without the KL refinement post-pass in Step 5. The data in Table 3 were derived from the same experimental tests described in Table 2, so the KL algorithm is given about 5% more time than the PHC/SG algorithm.) Perhaps surprisingly, the PHC/SG algorithm still manages to outperform the KL algorithm on five of the graphs (8870, PrimGA1, PrimGA2, bm1, and $U(1000, 0.04)$), substantially in some cases.

4 Conclusions

The PHC/SG+KL algorithm is undoubtedly an improvement over the KL algorithm, but it remains to be seen how effective it is relative to other recently reported algorithms that use explicit clustering heuristics. Furthermore, we can as yet offer no analysis that would

[6] Unfortunately a direct comparison with other algorithms on the MCNC graphs based on published figures is not currently possible, because the common convention is to report cut-set size in terms of nets (edges in a hypergraph) rather than edges in the graph derived from the original hypergraph, which is what we have done here for consistency with other presentations [1, 8, 10]. Furthermore, we bisect the graph on the basis of the number of vertices in each half of the bisection, not the weighted sum of the areas associated with them.

	KL		PHC/SG		
	avg min		avg min		% improvement
Graph	cut-set size	σ	cut-set size	σ	over KL
19ks	1020.8	33.5	1093.2	101.4	-7.1%
5655	603.2	4.3	612.8	2.9	-1.6%
8870	52.8	1.3	52.0	0.0	1.5%
PrimGA1	235.6	15.6	230.0	4.2	2.4%
PrimGA2	1051.6	77.5	751.8	31.1	28.5%
Test02	1172.8	11.9	1209.0	15.9	-3.1%
Test03	821.8	8.1	827.6	9.5	-0.7%
Test04	1191.2	2.2	1218.4	11.7	-2.3%
Test05	1887.4	26.4	1968.6	41.1	-4.3%
Test06	1194.4	3.6	1222.6	12.5	-2.4%
bm1	240.4	14.5	217.4	4.0	9.6%
$G(1000, 0.0025)$	98.2	3.0	98.4	1.1	-0.2%
$U(1000, 0.04)$	28.6	5.7	4.2	0.4	85.3%

Table 3: Kernighan-Lin and PHC/SG: an empirical comparison.

indicate why PHC/SG+KL is much better than KL on some graphs but not on others. Our agenda for future work therefore includes a thorough time-equated empirical comparison of the most promising clustering-based heuristics for graph bisection, including PHC/SG+KL, and an attempt to discover correlates between quantitative measures of a graph's structure and the performance of different algorithms.

Furthermore, we plan to generalize the PHC/SG+KL algorithm to other graph-partitioning problems. In commonly encountered problems of practical significance, more than two partitions are permitted, the requirement of exact equality of partition sizes is relaxed, and the vertices and edges are weighted. The simple nature of the seed-growth heuristic should allow for straightforward generalization to these cases, though its performance remains to be seen.

References

[1] T. Bui, C. Heigham, C. Jones, and T. Leighton. Improving the performance of the Kernighan-Lin and simulated annealing graph bisection algorithms. In *Proceedings of the 26th ACM/IEEE Design Automation Conference*, pages 775–778, 1989.

[2] J. Cong and M. Smith. A parallel bottom-up clustering algorithm with applications to circuit partitioning in VLSI design. In *Proceedings of the 30th ACM/IEEE Design Automation Conference*, pages 755–760, Dallas, TX, June 1993.

[3] W. E. Donath. Logic partitioning. In B. Preas and M. Lorenzetti, editors, *Physical Design Automation of VLSI Systems*, pages 65–86. Benjamin/Cummings, 1988.

[4] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitioning. In *Proceedings of the 19th Design Automation Conference*, pages 175–181, Las Vegas, NM, 1982.

[5] J. Garbers, H. J. Prömel, and A. Steger. Finding clusters in VLSI circuits. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, pages 520–523, Santa Clara, California, Nov. 1990.

[6] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York, 1989.

[8] M. K. Goldberg and M. Burstein. Heuristic improvement technique for bisection of VLSI networks. In *Proceedings of the IEEE International Conference on Computer Design*, pages 122–125, Port Chester, NY, 1983.

[9] L. Hagen and A. B. Kahng. A new approach to effective circuit clustering. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 422–427, Santa Clara, California, Nov. 1992.

[10] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning. *Operations Research*, 37(6):865–892, Nov.-Dec. 1989.

[11] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, Feb. 1970.

[12] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34:975–986, 1984.

[13] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[14] B. Krishnamurthy. An improved min-cut algorithm for partitioning VLSI networks. *IEEE Transactions on Computers*, C-33:438–446, 1984.

[15] T.-K. Ng, J. Oldfield, and V. Pitchumani. Improvements of a mincut partition algorithm. In *Proceedings of the IEEE International Conference on Computer Design*, pages 470–473, Santa Clara, CA, 1987.

[16] Y.-C. Wei and C.-K. Cheng. A two-level two-way partitioning algorithm. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, pages 516–519, Santa Clara, CA, Nov. 1990.

[17] Y.-C. Wei and C.-K. Cheng. Ratio cut partitioning for hierarchical design. *IEEE Transactions on Computer-Aided Design*, 10(7):911–921, July 1991.